





A PILOT COMPARATIVE ANALYSIS OF THE CUCKOO AND DRAKVUF SANDBOXES: AN END-USER PERSPECTIVE

Slaviša Ž. Ilić^a, Milan J. Gnjatović^b,
Brankica M. Popović^c, Nemanja D. Maček^d,

^a Ministry of Defense of the Republic of Serbia,
Belgrade, Republic of Serbia,
e-mail: slavisa.ilic@mod.gov.rs, **corresponding author**,
ORCID iD:  <https://orcid.org/0000-0003-3314-5794>

^b University of Criminal Investigation and Police Studies,
Belgrade, Republic of Serbia,
e-mail: milan.gnjatovic@kpu.edu.rs,
ORCID iD:  <https://orcid.org/0000-0002-0343-7596>

^c University of Criminal Investigation and Police Studies,
Belgrade, Republic of Serbia,
e-mail: brankica.popovic@kpu.edu.rs,
ORCID iD:  <https://orcid.org/0000-0002-8276-1481>

^d School of Electrical and Computer Engineering,
Academy of Technical and Art Applied Studies,
Belgrade, Republic of Serbia,
e-mail: nmacek@viser.edu.rs,
ORCID iD:  <https://orcid.org/0000-0002-3465-7524>

DOI: 10.5937/vojtehg70-36196; <https://doi.org/10.5937/vojtehg70-36196>

FIELD: Computer sciences, IT, Cyber security

ARTICLE TYPE: Original scientific paper

Abstract:

Introduction/purpose: This paper reports on a pilot comparative analysis of the Cuckoo and Drakvuf sandboxes. These sandboxes are selected as the subjects of the analysis because of their popularity in the professional community and their complementary approaches to analyzing malware behavior.

Methods: Both sandboxes were set up with basic configurations and confronted with the same set of malware samples. The evaluation was primarily conducted with respect to the question of to what extent a sandbox is helpful to the human analyst in malware analysis. Thus, only the information available in Web console reports was considered.

Results: Drakvuf is expected to perform better when confronted with evasive malware and so-called "file-less" malware. Although still not mature in terms of integration, customization and tools, this sandbox is considered

a second generation sandbox because of its agentless design. On the other hand, the Cuckoo sandbox creates a better overall experience: it is supported through good documentation and strong professional community, better integrated with various tools, support more virtualization, operating system and sample types, and generates more informative reports. Even with a smaller capacity to prevent evasive malware, its Python 2 agent script makes it more powerful than Drakvuf.

Conclusion: To achieve the optimal open-source sandbox-based protection, it is recommended to apply both the Cuckoo and Drakvuf sandboxes. In circumstances of limited resources, applying the Cuckoo sandbox is preferable, especially if exposure to malware deploying evading techniques is not frequently expected.

Key words: Sandbox, Cuckoo, Drakvuf, Malware behavior analysis.

Introduction

The number of malware attacks has recently increased significantly (e.g., it has been doubled in the period between 2015 and 2019 (Melvin & Kathrine, 2020) and the average time needed to detect a data breach is considerable (e.g., in 2020 it took 203 days in average), (IBM, 2020). The identification of highly sophisticated, target specific and stealthy operated cyber threats is a challenging task, because of their underlying characteristics such as encrypted covert communication, sophisticated attack techniques, continuous monitoring and control of victim's resources, wiping or masking the traces, etc. (Chakkaravarthy et al, 2019)

Due to the complexity and severity of advanced cyber threats, defenders of valuable assets aim at discovering threats before they get in a defensive perimeter. In line with this aim, this paper provides a pilot comparative analysis of two open-source and the most frequently used sandbox solutions: Cuckoo and Drakvuf.

Sandboxes and the experimental environment

A cybersecurity sandbox is a physical or virtual environment used to execute suspicious file samples or run programs without interfering with a monitoring system or permanently affecting a device they are running on (Arntz, 2020; Chakkaravarthy et al, 2019). The sandboxing is used to detect potentially malicious codes and applications before serving them up to critical devices (Arntz, 2020). The detection is based on malware behavior analysis, which may be roughly described by an analogy to biometric behavioral description (Tot et al, 2021).

A sandbox usually consists of a management part and virtual machines (VMs) which represent victim hosts. VMs are typically configured similarly to virtual and physical computers in a given organization in order to mimic the production environment which is being protected from malware attacks. When suspected files are executed in these VMs, it is possible to monitor their behavior and react before they occur in a production environment.

The Cuckoo and Drakvuf sandboxes are selected as the subjects of the analysis in this study because of their popularity in the professional community and their complementary approaches to analyzing malware behavior. Cuckoo uses a Python script-shaped agent in the analysis VM, while Drakvuf applies an agentless approach. The network architecture adopted in the reported study is shown in Figure 1.

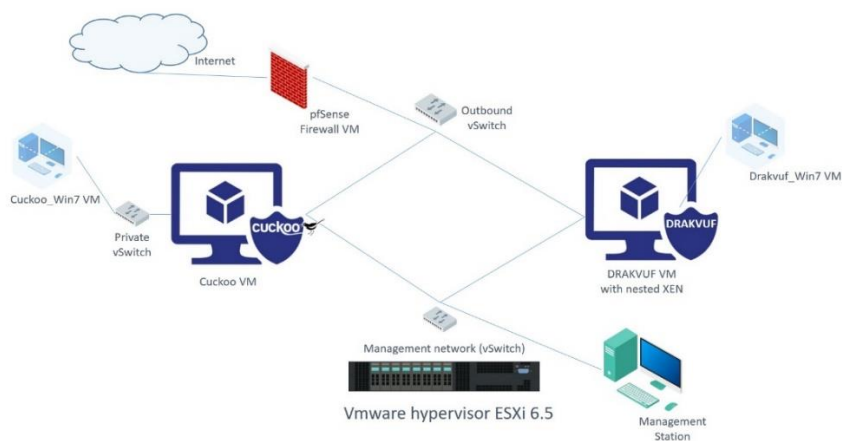


Figure 1 – ESXi network architecture with Cuckoo and Drakvuf virtual machines
 Рис. 1 – Сетевая архитектура ESXi с виртуальными машинами Cuckoo и Drakvuf
 Слика 1 – Мрежна архитектура ESXi са виртуелним машинама Куку и Драквуф

Both sandbox environments can be accessed from the management network for the purposes of configuration and submission of samples for analysis through a Web interface. The characteristics of individual VMs are provided in Table 1.

It is usual for Cuckoo and DRAKVUF to be installed on bare-metal (or in VM) and to have analysis VMs nested (i.e., Type 2 virtualization). In this study, we applied a different approach for Cuckoo since it supports VMware ESXi API calls. The ESXi communication with VMs allows for a more configurable environment and a more comprehensive analysis (e.g., by taking a snapshot and applying additional tools outside a sandbox

environment). In contrast, this approach is not feasible for Drakvuf, so the nested Xen virtualization is applied as the underlying hypervisor.

Table 1 – Virtual machines characteristics (operating system types, virtualization types and basic configuration)

Таблица 1 – Характеристики виртуальных машин (типы операционных систем, виды виртуализации и базовая конфигурация)

Табела 1 – Карактеристике виртуалних машина (типови оперативних система, типови виртуализације и основна конфигурација)

VM name	OS	Virt.	Basic configuration
Cuckoo	Ubuntu 18.04	ESXi	Version 2.05
Cuckoo_Win7	Windows7, 32-bit	ESXi	Office 2007, UAC and AV disabled
DRAKVUF	Ubuntu 20.04	ESXi	Version 0.18
Drakvuf_Win7	Windows7, 32-bit	Xen	Office 2007, UAC and AV disabled
pfSense	FreeBSD	ESXi	Open-source firewall

The pfSense firewall is configured in front of the whole environment in order to:

- prevent any traffic from leaving the experimental environment,
- provide additional real-time monitoring of network connections induced by the analyzed samples, and
- allow for keeping track of the sandbox-based traffic analysis across time.

A basic insight into the design of the sandboxes is provided below.

Cuckoo

The Cuckoo sandbox allows for dynamic detecting of runtime behaviors in an isolated environment, i.e. a virtual machine (including API calls, network traffic, files dropped, etc.) by the use of signatures, written as Python 2 scripts, that detect a broad range of malware, from a simple key logger to a more complicated execution of a process that has an injected code.

The malware detection is achieved via cuckoomon.dll, a dynamic link library injected into a process that allows for run-time logging of its behavior manifestations, which are then reported back to the main Cuckoo sandbox process.

Cuckoo may be integrated with local email solutions and intrusion prevention systems to identify ransomware and other potentially malicious entities, and to prevent potential breaches and data loss.

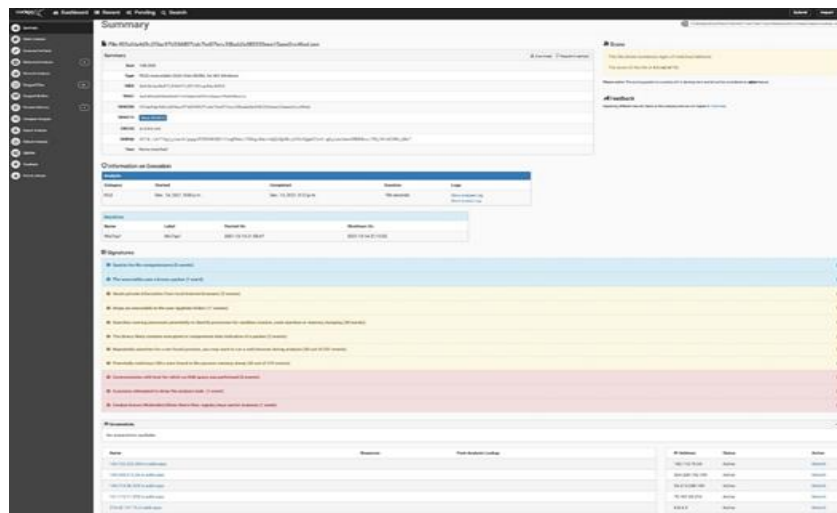


Figure 2 – Analysis results in the Cuckoo sandbox

Рис. 2 – Результаты анализа в песочнице Cuckoo

Слика 2 – Резултати анализе у софтверском окружењу Куку за изоловано извршавање програма

The analyst uses a Cuckoo host machine to manage the analysis through a command line or available Web interface. In the scope of an analysis, malware samples are submitted and reports are collected (Figure 2). Before malware execution, a VM's snapshot is reversed to the initial state which ensures that tracks of previous analyses do not interfere with the currently ongoing analysis. An in-guest Python agent serves to execute a malware sample and send a report back to the Cuckoo host.

Drakvuf

Drakvuf (Lengyel et al, 2014) is a VMI-based sandbox which has the ability to trace kernel-level and user-level malware (Melvin & Kathrine, 2020). VMI stands for Virtual Machine Introspection – external access to the virtual hardware state, which can monitor process execution, file operations, system calls and kernel function traces, all at the hypervisor level, with the ability to spot kernel rootkits and reduce the possibility for malware to use its evading techniques (Melvin & Kathrine, 2020). Instead of an in-guest agent, Drakvuf uses the breakpoint injection technique in which an instruction is written into the VM's memory at code locations of interest (Lengyel et al, 2014). By configuring a CPU to issue VMEXIT when breakpoints are executed, and configuring Xen (a virtualization hypervisor)

to forward these events to control domain, DRAKVUF is capable of trapping the execution of any code within the analysis machine. The #BP technique, previously used for the stealth debugging, is used for automatic execution tracing of the entire operating system, including also the Kernel internal functions.

Drakvuf's initial tests showed a great potential and the development of this sandbox was continued, providing a modern and powerful open-source malware analysis platform. The project is maintained and available at: <https://drakvuf.com>. In this study, we use the customized solution called Drakvuf sandbox (CERT of Poland) which is an actively developed project (Cert Polska, 2021, 2022), since it is easy to install and configure.



Figure 3 – Analysis results in the Drakvuf sandbox

Рис. 3 – Результаты анализа в песочнице Drakvuf

Слика 3 – Резултати анализе у софтверском окружењу Драквуф за изоловано извршавање програма

Analysis results

Since the considered sandboxes are not of the same type, it is challenging to introduce a metric for their comparison. Instead, we decided to use a descriptive approach to evaluate different features we consider relevant. The observed features are shown in Table 1, where sign “+” is assigned to the sandbox that performs better with respect to a given feature.

Table 2 – Analysis results
Таблица 2 – Результаты анализа
Табела 2 – Резултати анализе

	Feature	Cuckoo	Drakvuf
1	Complexity of installation and setup		+
2	Scalability		+
3	Reporting	+	
4	Execution time	+	
5	Supported file types	+	
6	Evasion prevention		+
7	Variety of analysis VM versions, underlying hardware and hypervisor support	+	
8	Integration with other tools and customization	+	
9	Automated samples submission and API	+	
10	Signatures (static analysis, PE, etc.)	+	
11	Visualization		+

Complexity of installation and setup

Both sandboxes are available for free under the General Public License (GNU GPLv3) and reasonably well documented. As a more mature solution, Cuckoo is more extensively documented, which allows for comparatively easier installation and configuration. In addition, this sandbox is used by many organizations, including CERT of Poland, CERT of Estonia (publicly available at <https://cuckoo.cert.ee>), Checkpoint, Avira, etc. (Estonian Information System Authority - RIA, 2017; Checkpoint Software Technologies LTD, 2019; Sick, 2014; CERT Polska, 2019), and the professional community is strong in terms of use, problem solving, customization and modifications. However, since Cuckoo is based on Python 2 (version 2.7 is used in this study), it suffers from some problems caused by dependencies on Python 3 packages. This fact prevented us from using the latest 2.07 Cuckoo version in this study, and led us to downgrade the hosting operating system from Ubuntu 20.04 to Ubuntu 18.04. The Drakvuf sandbox has good basic documentation and respectable community. The Drakvuf Sandbox (CERT of Poland edition) is enriched with a Web interface and additional plugins which create the user experience near to Cuckoo. However, with respect to the complexity of installation and setup, Drakvuf performs better, mainly due to the packet dependency problems in Cuckoo. A full rewrite of Cuckoo for Python 3 in

cooperation with the CERT of Estonia is announced and its availability for customization is expected to improve the Cuckoo's rating with respect to this feature.

Scalability

Although there are certain modifications of Cuckoo aimed at achieving the scalability for the Amazon Web Services (AWS Cloud), this sandbox is generally not easy scalable. In contrast to this, Drakvuf is easily scaled (i.e., command "draksetup scale n" accepts an input argument that represents the number of instances to be automatically configured and started for parallel samples execution).

Reporting

Table 3 provides a comparative overview of the content provided in the analysis reports of the Cuckoo and Drakvuf sandboxes.

Table 3 – Comparative overview of the content provided in the analysis reports of the Cuckoo and Drakvuf sandboxes

Таблица 3 – Сравнительный обзор содержимого, представленного в отчетах анализа песочниц Cuckoo и Drakvu

Табела 3 – Упоредни преглед садржаја извештаја добијених анализом у софтверским окружењима Куку и Дракувф за изоловано извршавање програма

Cuckoo	Drakvuf
Summary	Metadata
Information on execution	
Score (beta version functionality)	-
Yara	-
Sample download	-
Signatures (for three risk categories)	-
Screenshots	-
Network traffic information (available in a report)	Download network traffic (stored in a file)
Static analysis	-
Extracted Artifacts	-
Behavioral Analysis (more informative)	Process tree (more advanced graphical representation)
	Behavioral graph (more advanced graphical representation)
Dropped Files	-
Dropped Buffers	-
Process Memory	-
Additional operations (e.g., resubmit, reanalyze with reboot, etc.)	-

It could be observed that Drakvuf lacks many report features compared to Cuckoo, some of which could be derived from log files. The log files generated by both sandboxes are rather informative but not necessarily appropriate for human interpretation, and therefore we consider here only the information available in Web console reports.

However, the reporting functionality is primarily evaluated with respect to the question of to what extent a sandbox is helpful to the human analyst during the process of malware analysis. To assess this question, we have used 21 potential malware files, summarized in Table 4.

The samples are courtesy of the Virus total portal (Sood, 2021). The name of each sample (cf. the second column) is generated by taking the first 8 characters of its SHA-256 value.

The full 256-bit hashes are available but not provided because of possible misuse. The fourth column of Table 4 contains the numbers of antivirus engines that reported the sample as positive, while the fifth column contains the numbers of all antivirus engines that analyzed the sample. We introduce the following malware score to describe a sample (note that malware score will always be in range 0 to 1) as a division of number of positives (#P) by total number (#T).

The malware score of a sample is used as an external measure according to which the respective behavior reports obtained from the Cuckoo and Drakvuf sandboxes were evaluated. The details of this evaluation are given in Table 5.

Although Cuckoo had some difficulties analyzing malware samples that operate on a large number of files (i.e., too many files error) due to which multiple analysis restarts were required and the sandbox failed to produce reports for two samples in Table 5 (samples 002d7712 and 003add9c), the overall conclusion is that it provides more informative reports.

Table 4 – Malware samples used to compare reporting functionality of the sandboxes

Таблица 4 – Образцы вредоносных программ, используемые для сравнения функциональности отчетов в песочницах

Табела 4 – Узорци штетних програма коришћених за поређење извештаја у софтверским окружењима за изоловано извршавање програма

	Samples	File extension	#P	#T	Cuckoo score [0-10]	Cuckoo execution time (sec.)	Drakvuf default execution time (sec.)
1	000a46e1	Executables	53	67	0.6	196	600
2	898ccbcd		17	59	0	198	600
3	001a9515		50	65	0.8	197	600
4	002cca70		45	67	0.8	197	600
5	002d7712		58	68	0.6	196	600
6	003add9c		52	65	0.6	197	600
7	003afda4		50	66	4.6	196	600
8	00a67cc9	DLL files	33	57	1.2	902	600
9	00cb2289		49	56	1	307	600
10	00e02090		34	57	1	502	600
11	00f0d52f		40	56	0.6	328	600
12	9e184db7	HTML files	16	57	0	623	600
13	4db0f844	Microsoft Word documents	37	61	0	196	600
14	b56da6b0		38	60	0.4	214	600
15	e6871658		38	58	0	618	600
16	ea7db3d3	Microsoft Excel documents	29	61	0.4	196	600
17	72297378		43	59	0.4	197	600
18	6383c1aa		47	61	0.4	196	600
19	5c5d1602		18	59	0.4	196	600
20	0ecb0f42		30	55	0.4	196	600
21	0c0fe7f7		34	60	0.4	196	600

Table 5 – Details of report evaluation
 Таблица 5 – Детали оценки отчетов
 Табела 5 – Деталји вредновања извештаја

Sample	Malware score	Cuckoo report	Drakvuf report
000a46e1	0.79	Score 0.6. File operations and 2 file creation 000a46e1...exe and c:\ttdxptt.exe, various registry modifications.	More than 20 processes and files created (limited with execution time).
898cbcd	0.29	Score 0. Yara embedded_pe Contains an embedded PE32 file, embedded_win_api. A non-Windows executable contains win32 API functions names shellcode. Matched shellcode byte patterns. No behavioral analysis.	None.
001a9515	0.77	None.	Creation of 2 files and processes 001a9515.exe and HelpMe.exe
002cca70	0.67	Score 3.8. Yara vmdetect - Possibly employs anti-virtualization techniques. 160 files dropped, connection to unavailable IP address for which no DNS request was made, creates modified copy of itself, etc. Various traces in static analysis.	None.
002d7712	0.85	Could not be reported since the analysis results size is greater than supported.	Creation of 9 files (exe, dll, bcf) and 4 processes, 16 registry entries.
003add9c	0.80	Could not be reported since the analysis results size is greater than supported.	Creation of 16 files (dll, exe), 9 processes, around 100 registry entries.
003afda4	0.76	Score 7.8. HTTP requests, steals private information from Internet browser, drops executables, search running process, potentially malicious URLs, communication with host without DNS query, delay attempted, enumerates services, installs itself for auto-run, creates modified copies of itself, creates worm files, registry keys and generates ICMP traffic.	Creation of 10 files and processes.
00a67cc9	0.58	Score 2.2. One of more buffers contain embedded PE file. Indication of a packer. Various registry accesses.	Creation of regsvr32.exe processes.

Sample	Malware score	Cuckoo report	Drakvuf report
00cb 2289	0.88	Score 1.2. Various file creation, KERNELBASE.dll.mui etc, DLL's used, process rundll32.exe. Signatures: Indication of a packer - a section with a high entropy has been found.	Creation of regsvr32.exe processes. File creation regsvr32.exe-8461dbee.pf. Process ID 2288.
00e0 2090	0.60	Score 1. Indication of a packer. Potentially malicious URLs found in process memory.	Creation of regsvr32.exe processes. File creation regsvr32.exe-8461dbee.pf. Process ID 2352.
00f0 d52f	0.71	Score 2.2. Indication of a packer. HTTP request, communication with a host for which no DNS query was performed.	Creation of regsvr32.exe processes. File creation regsvr32.exe-8461dbee.pf Process ID 2284.
9e18 4db7	0.28	Score 1.8. Potentially malicious URLs in the process memory dump. Uses Windows utilities for basic Windows functionalities. Resumes thread in a remote process (potential process injection). Two files dropped.	Creates mshta.exe process with 13 threads which makes 17 registry entries in part of Internet connection settings.
4db0 f844	0.61	Score 9.8. Creation of 18 files. Powershell sending data, 6 http requests, potentially malicious URLs, URL downloaded by powershell script, winword.exe and powershell.exe wrote an executable to disk which then attempted to execute, powershell downloaded payload, etc.	About one hundred of registry accesses.
b56d a6b0	0.63	Score 11. Creation of 18 files. Communication with host without DNS query, a script was created with unexpected parent, potential payload download by powershell.exe non safe listed process created, and ICMP traffic.	About one hundred of registry accesses.
e687 1658	0.66	Score 7.4. Various file creation (DOC, LNK, scripts). Suspicious process creation. Malicious URLs found in memory dump. Extracted script. Dropped files which are executed.	Creation of one file, about one hundred of registry accesses in networking and Microsoft Office parts.
ea7d b3d3	0.48	Score 2.2. Yara: embedded_win_api - A non-Windows executable contains win32 API functions names. Communication with host without DNS query.	Hundreds of registry accesses.

Sample	Malware score	Cuckoo report	Drakvuf report
7229 7378	0.73	Score 2.2. Http requests, changes read-write memory protection to read-execute probably to avoid detection. Potentially malicious URLs, communication with host without DNS query.	Hundreds of registry accesses.
6383 c1aa	0.77	Score 1.8. Changes read-write memory protection to read-execute probably to avoid detection, potentially malicious URLs found, communication with host without DNS query.	Hundreds of registry accesses.
5c5d 1602	0.31	Score 1.8. Changes read-write memory protection to read-execute probably to avoid detection, communication with host without DNS query.	Hundreds of registry accesses.
0ecb 0f42	0.55	Score 3.2. HTTP requests, changes read-write memory protection to read-execute probably to avoid detection. Potentially malicious URLs, communication with host without DNS query. Connects to IP address that no longer responds to requests.	Hundreds of registry accesses.
0c0f e7f7	0.57	Score 3.0. HTTP requests, creates hidden or system file, changes read-write memory protection to read-execute probably to avoid detection. Potentially malicious URLs, communication with host without DNS query. Generates some ICMP traffic.	Hundreds of registry accesses.

Execution time

From Table 3, it can be observed that Cuckoo is more efficient for the given data. However, it should be noted that Drakvuf configuration supposes a constant execution time (default is 10 minutes, but could be lowered). The Drakvuf authors probably introduced this unbalanced trade-off between the efficacy and security in order to reduce the possibility for a sample to evade the sandbox environment.

Supported file types

The file types supported by the sandboxes are shown in Figure 4. The Cuckoo sandbox has a huge advantage in terms of supported file types (including various scripts, PDF and ZIP-file type extensions) and can be

customized for generic packages by selecting applications to handle a particular sample type.

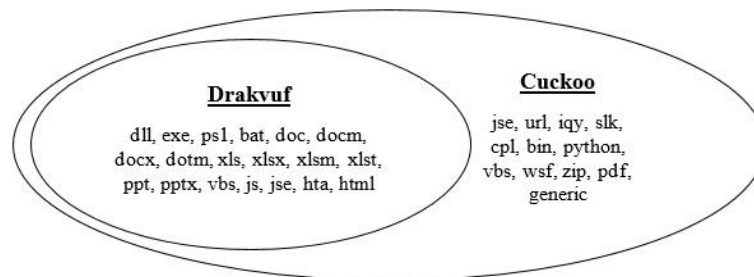


Figure 4 – Supported file types in Cuckoo and Drakvuf

Рис. 4 – Поддерживаемые типы файлов в безагентных песочницах Cuckoo и Drakvuf

Слика 4 – Подржани типови фајлова у софтверским окружењима Куку и Драквуф за изоловано извршавање програма

Evasion prevention

Since malware proved as evasive was not available in this study, the insight with respect to this feature is based on the sandboxes design and available research (Laing, 2017; Mills & Legg, 2021; Ferrand, 2015; Lengyel et al, 2014). It may be concluded that out-of-the-box Drakvuf performs better when confronted with evading malware, although both systems could be hardened to increase the probability of executing and reporting evasive malware.

Virtual machine, hypervisor and hardware support

Cuckoo supports Windows XP, Windows 7 64-bit and Windows 10 64-bit (not fully functional), Ubuntu 18.04 as a Linux guest operating system, and it can be configured to analyze samples in an Android environment under the Linux guest operating system. The configuration of Physical machine is also possible for the purpose of analysis. Drakvuf supports Windows 7-8, both 32-bit and 64-bit versions, 64-bit version of Windows 10 as well as 32-bit and 64-bit Linux systems running kernel 2.6.x and above, while the particular Drakvuf sandbox considered in this study is limited to Windows 7 (64-bit) and Windows 10 (64-bit) experimental.

Due to its design, Drakvuf is limited to the Intel processors, while Cuckoo can run without hardware limitations. The hypervisor support is also on the side of Cuckoo because it can communicate with analysis VMs under Xen, KVM, VMware ESXi, Oracle Virtual box, and almost on any other platform, while in the Drakvuf environment only Xen is natively

supported (KVM is in the experimental phase and VMware Workstation needs to be additionally configured). Due to its design, Drakvuf only supports nested virtualization. Thus, Cuckoo is also more advanced with respect to this feature, although Drakvuf provides enough options to work just fine in most environments.

Integration with other tools and customization

Since Cuckoo is a more mature solution, its integration possibilities are greater. Thus, it can be integrated with a range of tools for additional analysis (e.g., Cuckoo-droid, Signature updates, YARA rules, Suricata, Snort, Moloch, Volatility, Virustotal integration, etc) (Ashby, 2015; Checkpoint Software Technologies LTD, 2015). The Drakvuf sandbox allows integration with certain tools (e.g., Volatility and procmon for behavioral graph induction), but its integration possibilities are still significantly lower.

Automated samples submission and API

Cuckoo supports multiple samples submission, which in conjunction with its efficient execution allows a real-time analysis in environments with limited resources and may be applied to analyze large amounts of malware. REST API is implemented and easily accessible by the execution of a single command, enabling the automation of the analysis process. Drakvuf does support multiple samples submission but its API is still undocumented and we could not find the way to effectively use it.

Signatures

Signatures are probably the most lacking feature in Drakvuf. In contrast, Python scripts in Cuckoo are automatically updated from the repository and create signatures that recognize malicious behavior of samples. YARA rules can be defined and applied to improve this process. Signatures are also applied in the static analysis of samples.

Visualization

Cuckoo has a beta version scoring system which is visually very illustrative, but not fully informative for detailed analyses in which visually advanced reports with signatures are substantially useful. In Drakvuf, the Process tree and Behavioral graph are very useful visual tools which make Drakvuf a slightly more advanced solution with respect to the visualization functionality.

Conclusion

Table 2 shows that the Cuckoo sandbox performs better with respect to many features. However, the answer to the question of which sandbox to apply depends on the expected malware behavior.

Drakvuf is expected to perform better when confronted with evasive malware and so-called “file-less” malware (residing only in the RAM of a device). It is also suitable for capturing traces that a malware attempts to clean (i.e., deletion of temporary files), since it fetches deleted files by intercepting internal kernel calls related to the file deletion operations. On the other hand, Drakvuf has its limitations including the use of the injection mechanism to automatically start a sample (Lengyel et al, 2014) which a malware can exploit to evade an abnormal start, but research demonstrates the potential of this sandbox with respect to evading malware techniques. Although still not mature in terms of integration, customization and tools, it is considered a second generation sandbox because of its agentless design (Laing, 2017; Richards, 2021; Lengyel et al, 2014)

The Cuckoo sandbox creates a better overall experience: it is supported through good documentation and strong professional community, better integrated with various tools, supports more virtualization, operating systems and sample types. With the Python 3 rewrite, Cuckoo 3 (Hatching International B.V., 2022) is expected to perform even better. Even with a smaller capacity to prevent a malware to evade the sandbox environment, its Python 2 agent script makes it more powerful than Drakvuf. Recent research including 539 organizations and companies in Europe and USA (Spiceworks, 2019) shows that 92% of the companies apply server virtualization solutions, and predicts that the increasing number of VMs in production environments could result in lowering the frequency of evasion techniques since attackers probably would not allow to be deprived of the opportunities to target these machines.

At the given point, to achieve an adequate or optimal open-source sandbox-based protection and improve cyber security risk management practices (Ilić, 2012), it is recommendable to apply both the Cuckoo and Drakvuf sandboxes. In circumstances of limited resources, applying the Cuckoo sandbox is preferable, especially if exposure to malware deploying evading techniques is not frequently expected.

References

- Arntz, P. 2020. Sandbox in security: what is it, and how it relates to malware. *Malwarebytes LABS blog*, 24 September [online]. Available at: <https://blog.malwarebytes.com/awareness/2020/09/sandbox-in-security> [Accessed: 30 January 2022].
- Ashby, C. 2015. Extending Cuckoo Framework. *PenTest magazine*, 12 March [online]. Available at: <https://pentestmag.com/cuckoo>. [Accessed: 30 January 2022].
- CERT Polska. 2019. Strengthening our malware analysis capabilities. *Official web site of CERT Polska (part of NASK)*, 21 February [online]. Available at: <https://cert.pl/en/posts/2019/02/strengthening-our-malware-analysis-capabilities/> [Accessed: 30 January 2022].
- CERT Polska. 2021. DRAKVUF Sandbox (v0.18.1). *Official repository of the DRAKVUF Sandbox project*, 28 October [online]. Available at: <https://github.com/CERT-Polska/drakvuf-sandbox/releases/tag/v0.18.1> [Accessed: 30 January 2022].
- CERT Polska. 2022. DRAKVUF Sandbox Documentation. *DRAKVUF Sandbox documentation at Read the docs*, 10 February [online]. Available at: https://drakvuf-sandbox.readthedocs.io/_/downloads/en/latest/pdf. [Accessed: 10 February 2022].
- Chakkaravarthy, S.S., Sangeetha, D. & Vaidehi, V. 2019. A Survey on malware analysis and mitigation techniques. *Computer Science Review*, 32, pp.1-23. Available at: <https://doi.org/10.1016/j.cosrev.2019.01.002>.
- Checkpoint Software Technologies LTD. 2015. CuckooDroid Book, Revision 13502746. *CuckooDroid at Read the docs* [online]. Available at: <https://cuckoo-droid.readthedocs.io/en/latest> [Accessed: 10 February 2022].
- Checkpoint Software Technologies LTD. 2019. Cuckoo SandBox on AWS. *Checkpoint research*, 11 March [online]. Available at: <https://research.checkpoint.com/2019/cuckoo-system-on-aws/> [Accessed: 10 February 2022].
- Estonian Information System Authority (RIA). 2017. Annual Cyber Security Assessment 2017. *Estonian Information System Authority (RIA) official website* [online]. Available at: https://www.ria.ee/sites/default/files/content-editors/kuberturve/ria_csa_2017.pdf [Accessed: 30 January 2022].
- Ferrand, O. 2015. How to detect the Cuckoo Sandbox and to Strengthen it? *Journal of Computer Virology and Hacking Techniques*, 11, pp.51-58. Available at: <https://doi.org/10.1007/s11416-014-0224-9>.
- Hatching International B.V., 2022. We know cuckoo. *Official web site of the Cuckoo developers* [online]. Available at: <https://hatching.io/cuckoo> [Accessed: 30 January 2022].
- IBM Corporation. 2020. IBM Security, report: IBM Cost of a Data Breach Report. *IBM official web site*. July [online]. Available after registration at: <https://www.ibm.com/security/digital-assets/cost-data-breach-report> [Accessed: 30 January 2022].

Ilić, S. 2012. CLOUD COMPUTING - Information assurance aspects in government use. In: *Proceedings of XVIII conference YU INFO*, Kopaonik, Serbia, March 01-03.

Laing, B. 2017. First-generation sandbox solutions do not beat evasive malware. *IDG Connect*. 8 February [online]. Available at: <https://www.idgconnect.com/article/3581202/first-generation-sandbox-solutions-do-not-beat-evasive-malware.html> [Accessed: 10 February 2022].

Lengyel, T.K., Maresca, s., Payne, B.D., Webster, G.D., Vogl, S. & Kiayias, A. 2014. Scalability, fidelity and stealth in the DRAKVUF dynamic malware analysis system. In: *ACSAC '14: Proceedings of the 30th Annual Computer Security Applications Conference*, New York, NY, USA, pp.386-395, December. Available at: <https://doi.org/10.1145/2664243.2664252>.

Melvin, A.A.R. & Kathrine, G.J.W. 2020. Quest for Best: A Detailed Comparison between Drakvuf - VMI-Based and Cuckoo Sandbox-Based Technique for Dynamic Malware Analysis. In: *Peter, J., Fernandes, S. & Alavi, A. (Eds.) Intelligence in Big Data Technologies - Beyond the Hype. Advances in Intelligent Systems and Computing*, 1167. Springer, Singapore. Available at: https://doi.org/10.1007/978-981-15-5285-4_27.

Mills, A. & Legg, P. 2021. Investigating Anti-Evasion Malware Triggers Using Automated Sandbox Reconfiguration Techniques. *Journal of Cybersecurity and Privacy*, 1, pp.19-39. Available at: <https://doi.org/10.20944/preprints202010.0305.v1>.

Richards, K. 2021. VMRay – The Hypervisor-Based Sandbox That Cannot be Detected (interview with Carsten Willems). *VpnMentor* [online]. Available at: <https://www.vpnmentor.com/blog/vmray-hypervisor-based-sandbox-not-detected>. [Accessed: 30 January 2022].

Sick, T. 2014. Cuckoo Sandbox vs. Reality. *Avira official web site*, 11 November [online]. Available at: <https://www.avira.com/en/blog/cuckoo-sandbox-vs-reality-2> [Accessed: 10 February 2022].

Sood, G. 2021. Virustotal: R Client for the virustotal API. R package version 0.2.2. *Virus total web portal* [online]. Available at: <https://www.virustotal.com> [Accessed: 10 February 2022].

-Spiceworks. 2019. The 2020 State of Virtualization Technology, Survey on 539 organizations and companies in Europe and USA. *Spiceworks* [online]. Available at: <https://www.spiceworks.com/marketing/reports/state-of-virtualization> [Accessed: 30 January 2022].

Tot, I.A., Bajčetić, J.B., Jovanović, B.Ž., Trikoš, M.B., Bogičević, D.Lj. & Gajić, T.M. Biometric standards and methods. *Vojnotehnički glasnik/Military Technical Courier*, 69(4), pp.963-977. Available at: <https://doi.org/10.5937/vojtahg69-32296>.

ЭКСПЕРИМЕНТАЛЬНЫЙ СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПЕСОЧНИЦ CISCOO И DRAKVUF: ВЗГЛЯД КОНЕЧНОГО ПОЛЬЗОВАТЕЛЯ

Славиша Ж. Илич^а, Милан Й. Гнятович^б,
Бранкица М. Попович^б, Неманя Д. Мачек^в

^а Министерство обороны Республики Сербия,
г. Белград, Республика Сербия, **корреспондент**

^б Университет криминалистики и полицейской подготовки,
г. Белград, Республика Сербия

^в Школа электротехники и вычислительной техники,
Академия технических и художественных прикладных исследований,
г. Белград, Республика Сербия

РУБРИКА ГРНТИ: 20.23.25 Информационные системы с базами знаний,
81.93.29 Информационная безопасность. Защита
информации

ВИД СТАТЬИ: оригинальная научная статья

Резюме:

Введение/цель: В данной статье представлен экспериментальный сравнительный анализ программных сред песочниц Ciscoo и Drakvuf. Эти системы были выбраны в качестве предмета анализа из-за их популярности в профессиональном сообществе и их взаимодополняющих подходов к анализу воздействия вредоносных программ.

Методы: Обе системы имеют базовые настройки и подвергаются воздействию одного и того же набора вредоносных программ. Анализ преимущественно проводился с целью выявления степени полезности песочниц для аналитика-человека при анализе вредоносных программ. Следовательно, учитывалась только та информация, которая была доступна в отчетах веб-интерфейсов наблюдаемых систем.

Результаты: Можно ожидать, что Drakvuf даст лучшие результаты при воздействии вредоносных программ, использующих методы обхода песочниц в виртуальных средах. Несмотря на то, что данная среда пока не достигла своей полной мощности относительно интеграции, настроек и доступных программных инструментов, ее все-таки можно считать представителем второго поколения изолированных систем программной среды, благодаря ее безагентной технологии. С другой стороны, песочница Ciscoo в целом более удобна для пользователей: она поддерживается хорошей документацией и сильным профессиональным сообществом, лучше интегрирована с различными программными

инструментами, поддерживает больше видов виртуализации, типов операционных систем и образцов, в том числе она лучше генерирует отчеты. Несмотря на то, что у этой песочницы гораздо меньше возможностей предотвращения атак вредоносных программ в виртуальной среде, применение сценария с выявлением действий вредоносных программ делает эту песочницу более эффективной.

Выводы: Для достижения оптимальной защиты на основе песочницы с открытым исходным кодом рекомендуется применять как песочницы Cuckoo, так и Drakvuf. В условиях ограниченных ресурсов предпочтительнее применять песочницу Cuckoo, особенно если не предполагается частое воздействие вредоносных программ, использующих метод уклонения от обнаружения.

Ключевые слова: изолированный запуск программы, Cuckoo, Drakvuf, динамический анализ вредоносных программ.

УПОРЕДНА ПИЛОТ-АНАЛИЗА СОФТВЕРСКИХ ОКРУЖЕЊА КУКУ И ДРАКВУФ ЗА ИЗОЛОВАНО ИЗВРШАВАЊЕ ПРОГРАМА: ПЕРСПЕКТИВА КРАЈЊЕГ КОРИСНИКА

Славиша Ж. Илић^а, Милан Ј. Ђатовић^б,
Бранкица М. Поповић^б, Немања Д. Мачек^в

^а Министарство одбране Републике Србије,
Београд, Република Србија, **аутор за преписку**

^б Криминалистико-полицијски универзитет, Београд, Република Србија

^в Висока школа електротехнике и рачунарства струковних студија,
Академија техничко-уметничких струковних студија,
Београд, Република Србија

ОБЛАСТ: рачунарске науке, информационе технологије,
информациона безбедност

ВРСТА ЧЛАНКА: оригинални научни рад

Сажетак:

Увод/циљ: У раду се приказује упоредна пилот-анализа софтверских окружења Куку и Драквуф за изоловано извршавање програма. Ови системи одабрани су за предмет анализе због своје популарности у професионалној заједници и комплементарних приступа анализи понашања злонамерних програма.

Метод: Оба система постављена су на основна подешавања и изложена истом скупу злонамерних програма. Анализа је примарно урађена с аспекта процене степена информативности добијених извештаја о извршавању злонамерних програма за људског аналитичара. Стога су, као предмет анализе, узете у обзир само информације доступне у веб-интерфејсима посматраних система.

Резултати: Може се очекивати да ће Драквуф остварити бољи учинак када се изложи злонамерним програмима који примењују технике избегавања извршавања у виртуалним окружењима. Иако ово окружење још увек није остварило пун капацитет у смислу интегрисања, прилагођавања и доступних софтверских алата, може се сматрати представником друге генерације система за изоловано извршавање програма, због свог дизајна који искључује примену софтверског агента. С друге стране, окружење Куку ствара боље свеукупно корисничко искуство: подржано је добром документацијом и јаком професионалном заједницом, боље је интегрисано са различитим софтверским алатима, подржава више типова виртуелизације, оперативних система и типова узорака и генерише информативније извештаје. Иако поседује мањи капацитет за откривање злонамерних програма који примењују технике избегавања извршавања у виртуалним окружењима, могућност примене скрипти с дефиницијама злонамерног понашања програма чини ово окружење ефективнијим.

Закључак: Да би се постигла оптимална заштита, заснована на окружењима отвореног кода за изоловано извршавање програма, препоручује се примена оба разматрана система. У условима ограничених ресурса, примена система Куку пожељнија је, посебно ако се не очекује често излагање злонамерним програмима који примењују технике избегавања извршавања у виртуалним окружењима.

Кључне речи: изоловано извршавање програма, Ciscoo, Drakvuf, динамичка анализа злонамерних програма.

Paper received on / Дата получения работы / Датум пријема чланка: 01.02.2022.

Manuscript corrections submitted on / Дата получения исправленной версии работы / Датум достављања исправки рукописа: 16.03.2022.

Paper accepted for publishing on / Дата окончательного согласования работы / Датум коначног прихватања чланка за објављивање: 18.03.2022.

© 2022 The Authors. Published by Vojnotehnički glasnik / Military Technical Courier (www.vtg.mod.gov.rs, втг.мо.упр.срб). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/rs/>).

© 2022 Авторы. Опубликовано в «Военно-технический вестник / Vojnotehnički glasnik / Military Technical Courier» (www.vtg.mod.gov.rs, втг.мо.упр.срб). Данная статья в открытом доступе и распространяется в соответствии с лицензией «Creative Commons» (<http://creativecommons.org/licenses/by/3.0/rs/>).

© 2022 Аутори. Објавио Војнотехнички гласник / Vojnotehnički glasnik / Military Technical Courier (www.vtg.mod.gov.rs, втг.мо.упр.срб). Ово је чланак отвореног приступа и дистрибуира се у складу са Creative Commons лиценцом (<http://creativecommons.org/licenses/by/3.0/rs/>).

