DOI: https://doi.org/10.15276/hait.02.2021.5
UDC 658 + 512.011 + 681.326 + 519.713

Models and methods for diagnosing Zero-Day threats in cyberspace

Oleksandr S. Saprykin
ORCID: https://orcid.org/0000-0001-5399-5054: sashasapr@gmail.com
Kharkiv National University of Radio Electronics,14, Nauky Ave. Kharkiv, 61166, Ukraine

ABSTRACT

The article is devoted to the development of models and methods for detecting Zero-Day threats in cyberspace to improve the efficiency of detecting high-level malicious complexes that are using polymorphic mutators. The method for detecting samples by antivirus solutions using a public and local multiscanner is proposed. The method for diagnosing polymorphic malware using Yara rules is being developed. The multicomponent service that allows organizing a free malware analysis solution with a hybrid deployment architecture in public and private clouds is described. The cloud service for detecting malware based on open-source sandboxes and MAS, allowing horizontal scalability in hybrid clouds, and showing high capacity during malicious and non-malicious object processing is designed. The main task of the service is to collect artifacts after dynamic and static object analysis to detect zero-day threats. The effectiveness of the proposed solutions is shown. Scientific novelty and originality consist in the creation of the following methods: 1) detecting the sample by preinstalled antivirus solutions that allow static scanning in separate threads without requests restrictions for increasing the malware processing speed and restrict public access to confidential files; 2) diagnosing polymorphic malware using Yara rules, that allows detecting new modifications that are not detected by available solutions. The proposed hybrid system architecture allows to perform a retrospective search by families, tracking changes in destructive components, collect the malicious URLs database to block traffic to C&C servers, collect dropped and downloaded files, analyze phishing emails attachments, integrate with SIEM, IDS, IPS, antiphishing and Honeypot systems, improve the quality of the SOC analyst, decrease the incidents response times and block new threats that are not detected by available antivirus solutions. The practical significance of the results is in the cloud service development that combines MAS Sandbox and a modified distributed Cuckoo sandbox, which allows to respond to Zero-Day threats quickly, store a knowledge base for artifacts correlation between polymorphic malware samples, actively search for new malware samples and integrate with cyber protection hardware and software systems that support the Cuckoo API.

Keywords: Zero-Day Threats; Sandbox; Cloud Service; Antivirus; Diagnostics; Detection, Malware; Polymorphic Codes, Scanner, Emulator.

***For citation*: Saprykin O. S. Models and Methods for Diagnosing Zero-Day Threats in Cyberspace. *Herald of Advanced Information Technology*. 2021; Vol.4 No.2: 155-167. DOI: https://doi.org/10.15276 / hait.02.2021.5**

INTRODUCTION. PROBLEM STATEMENT

The ensuring security problem of cyberspace from Zero-Day threats is becoming more and more urgent every year. This form of cyber threat is the most high-tech and difficult to detect, as it occurs before the emergence of protective mechanisms from the security vendors. Developers of malicious programs are increasingly using unknown vulnerabilities in applications and operating systems to proceed to be undetected for a long time. It often takes a long time to write fixes for such threats, and the system is left unprotected until the zero-day vulnerabilities would be properly patched. According to WatchGuard [1], Zero-Day programs appear almost everywhere increase about 50 % of the total number of all malicious programs [2], which is the highest indicator in history (Fig. 1).

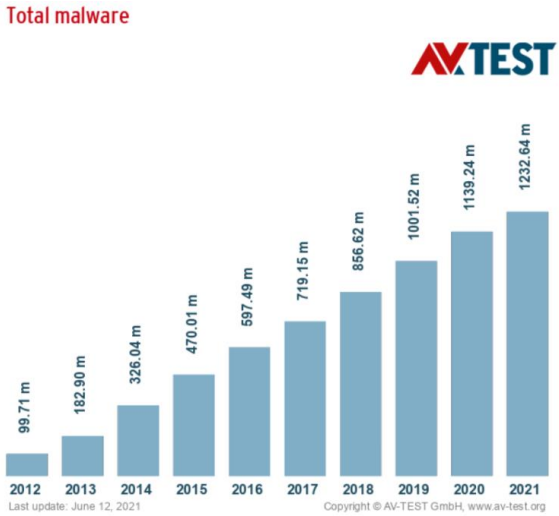


Fig. 1. **Annual detection of new malware around the world**

Source:
https://www.avtest.org/typo3temp/avtestreports/total_distribution_10-years_en.png?1623515107

The automatic analysis systems based on sandboxes that allow assessing the software malicious indicators by running and analyzing it in an isolated environment help to solve the problem of identifying previously unknown malicious samples.

© Saprykin O., 2021
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/deed.uk)

Instead of using signature-based methods to search for malicious activity, the sandbox analyzes the actions of the program in a virtual or real environment using the various operating systems. The sandbox allows to reproduce the actions of malicious programs in order to check how a virus will behave in various operating systems without any risks for real infrastructure.

LITERATURE OVERVIEW

Many free and paid sandbox solutions include various technologies for static and dynamic analysis. The Global Sandboxing Market Size By Component, By Vertical, By Geographic Scope And Forecast study predicts that by 2027 the market volume will reach 7.74 billion US dollars, an average annual increase of 12.65 % [3]. The solutions [13, 23] allow to use the neural network methods for detecting malicious programs, based on the sequence of function calls and their arguments how the input neural network training set. However, modern malicious programs can detect the cuckoo-mon.dll library injection to control API calls and terminate their work within the sandbox environment to prevent dynamic analysis of their executable file, the Yara rules detection method is used [4], that does not interact with the malicious program in any way. The method uses malware artifacts only [10]. Unitary coded qubit-matrix models, data structures, computing architectures and methods for parallel logical analysis of malware codes in cyber-physical space were proposed in [24]. New models, methods, blockchaine infrastructure for improvement of security, and also problems, their origins and solutions to protect cyberspace are presented in [19, 20], [21, 22], [23, 24], [25]. The article [26] describes a possibility to apply a sandboxing method within a cloud environment to enforce security perimeter of the cloud. The growing number of cyberattacks makes a significant contribution to the development of sandbox-based malware analysis systems. At the same time, the attackers are constantly looking for ways to detect the Sandbox environment to block the execution of destructive code and prevent it from running in a virtual environment. This in turn the sandbox makers force to develop technologies for hiding the Sandbox environment. Graph-based malware detection methods must build a behavior graph for each known malware, and they are difficult to apply in practice [27, 28].

THE AIM AND OBJECTIVES OF THE RESEARCH

The aim of the research is to significantly increase the efficiency of detecting complex malicious programs that are using polymorphic mutators via using the developed models and methods for Zero-Day threats diagnosing in cyberspace.

Research objectives: 1) Develop a method for detecting samples by antivirus solutions using a public and local multiscanner. 2) Develop a method for detecting polymorphic malware samples using Yara rules. 3) Design a cloud-based malware detection service based on open-source sandboxes and MAS that allows horizontal scaling in hybrid clouds and demonstrates high throughput during the malicious and non-malicious object processing. The main task of the service is to collect artifacts after dynamic and static object analysis to detect Zero-Day threats.

MAS MALWARE ANALYSIS CLOUD SERVICE

To solve the polymorphic malware detections problems the cloud service is proposed that includes three main modules:

1) The server controls the input and output data flow, performs a static analysis for the objects, launch scans using Yara rules [4] and Virustotal service [5], collects artifacts after dynamic analysis, scans the memory and traffic dumps using Yara rules and generates a final verdict on the presence of a destructive component. All collected artifacts are saved to the MySQL database (Fig. 2) [6].

2) Multiscanner allows to scan a file using preinstalled antivirus solution and preconfigured do not send samples for analysis to antivirus laboratories, therefore allowing to scan sensitive enterprise files that cannot be sent to public multiscanners.

3) Sandbox is a virtual or real environment with a preinstalled operating system that includes the agent for launching malicious programs and plugins for dynamic analysis. Plugins collect memory dumps o new running processes in the system and dumps of injects in the processes, network traffic, file, and Windows registry activities, find anomalies in the Windows host file, catch anomalies in the system to detect user-mode and kernel-mode rootkits, collect information about created mutexes in the system, make screenshots of the malware's top window.

The main components of the MAS malware analysis service are shown in Fig. 3.

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

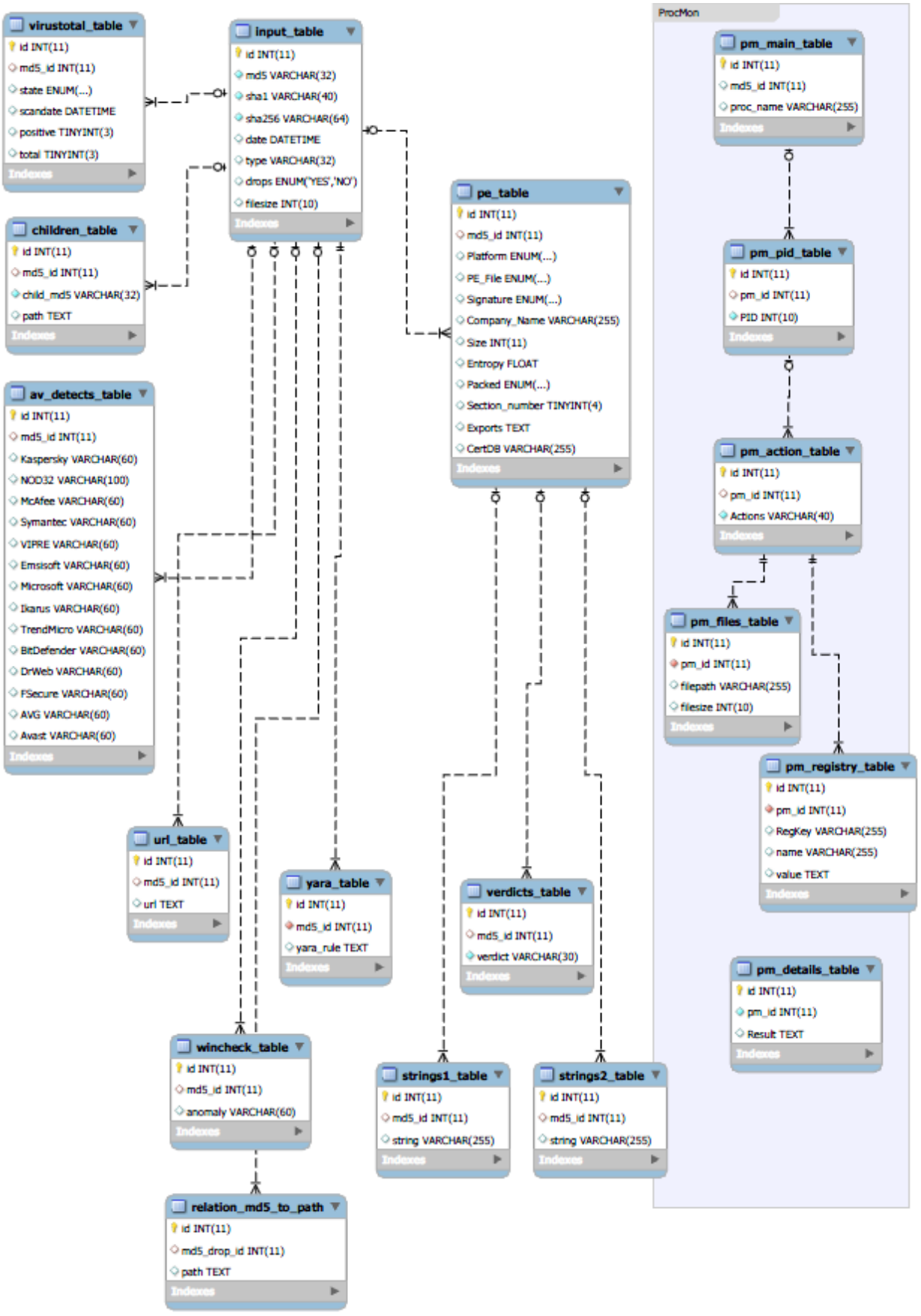


Fig. 2. **MAS cloud service database architecture**

Source: **Compiled by the author**

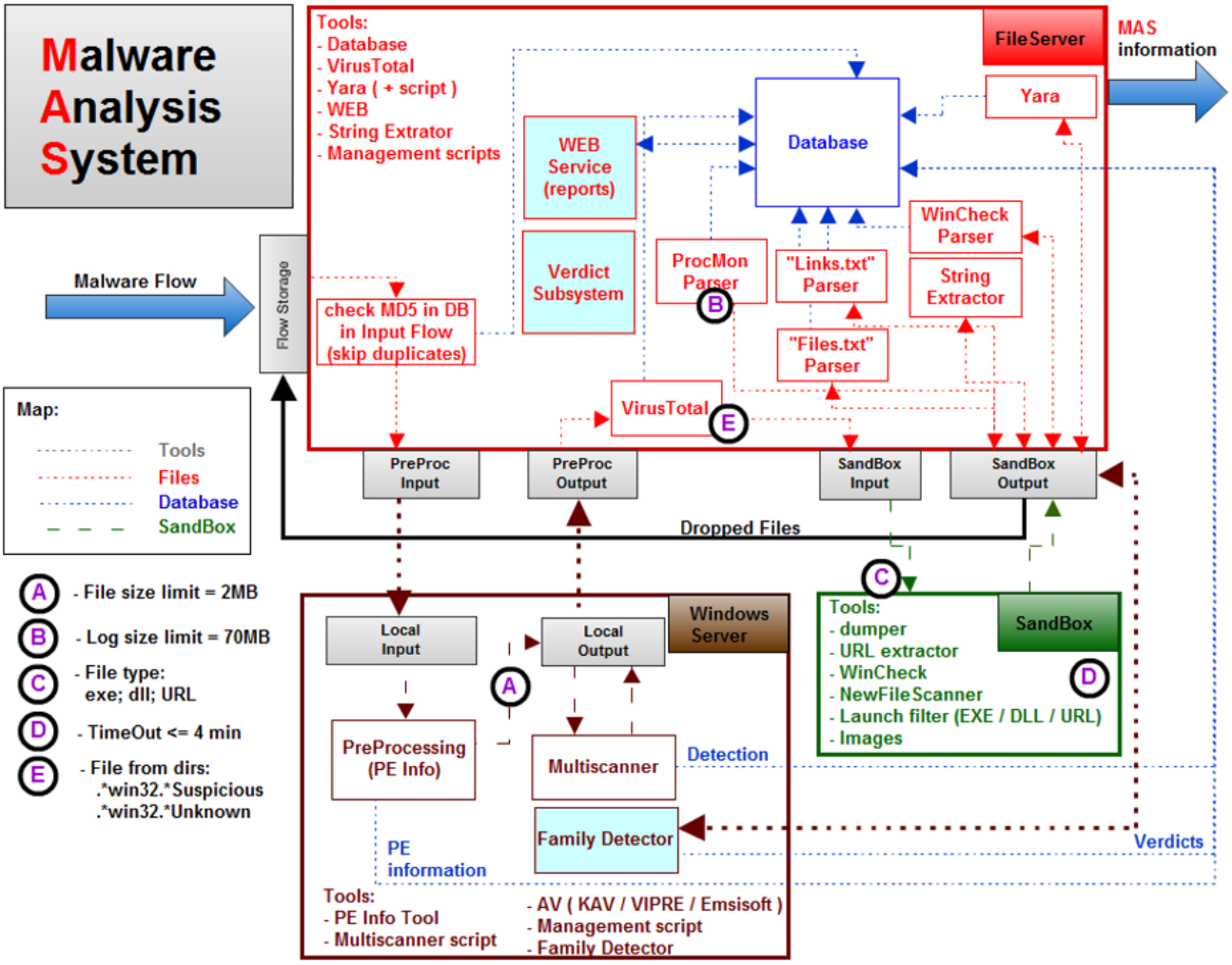


Fig. 3. **The main components of the MAS cloud service**
Source: **Compiled by the author**

The sandbox can use the Internet emulator "Inetsim" if a direct Internet connection is blocking malicious traffic on the ISP's side. If access to the Internet exists it allows to detect of anomalies in network traffic because of Suricata intrusion detection system [7] is installed on the server.

The main advantage of this module is the agent and plugins can be launched within the isolated real environment. The special sandbox instance can be created to analyze high-level malware that prevents its execution within the virtual environment to bypass detection mechanisms. The module allows you to configure a hybrid architecture based on real and virtual machines, both in a public cloud and in a private one. In this case, the real machine is placed in a completely isolated VLAN, and the connection with the cloud is configured through the Site-to-Site VPN.

The cloud service has a web interface (Fig. 4) with the ability to send a file for analysis, search for a malware report by the hash (Fig. 5), and search for malware families using specified artifacts.



Fig. 4. **The web interface of the cloud service MAS**
Source: **Compiled by the author**

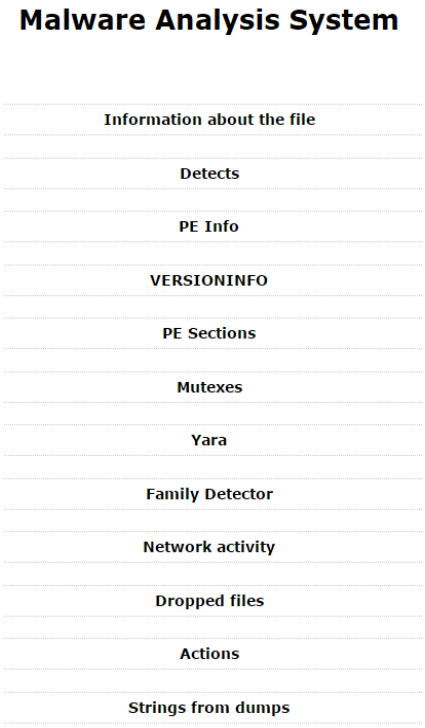


Fig. 5. **MAS modules**
Source: **Compiled by the author**

During the search, the correlation between the zero-day artifacts of the malicious program can be performed (Fig. 6 and Fig.7) to understand which file dropped or downloaded from the malicious file, find the CnC servers and search for Windows hooks.

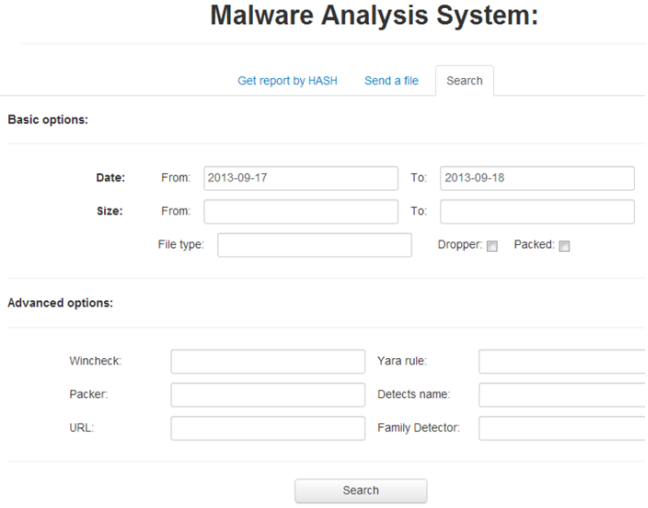


Fig. 6. **Panel for searching malicious objects by artifacts**

Source: **Compiled by the author**

#	MD5	File Type	Filesize	Yara Rule	Family Detector
1	a828a97d010b619bd9c677e2eba6d2ae	PE32	302592	BankerGeneric GenericInjector GenericBootkit Shiz	Backdoor.Win32.Shiz : 100
2	cb52fa0634a46245e4367b783d02c1d6	PE32	343040	BankerGeneric GenericInjector GenericBootkit Shiz	Backdoor.Win32.Shiz : 100
3	034d39867e3c71e27653051387a23605	PE32	314368	BankerGeneric GenericInjector GenericBootkit Shiz	Backdoor.Win32.Shiz : 100
4	2748026d60875711592d718d1fac95a9	PE32	322048	BankerGeneric GenericInjector GenericBootkit Shiz	Backdoor.Win32.Shiz : 100
5	7f827ce13a586391961bbb0404644681	PE32	335360	BankerGeneric GenericInjector GenericBootkit Shiz	Backdoor.Win32.Shiz : 99
6	a72bb4b3c9717865536ae93c4114b062	PE32	251392	BankerGeneric GenericInjector GenericBootkit Shiz	Backdoor.Win32.Shiz : 100

Fig. 7. **The search results of the cloud MAS**
Source: **Compiled by the author**

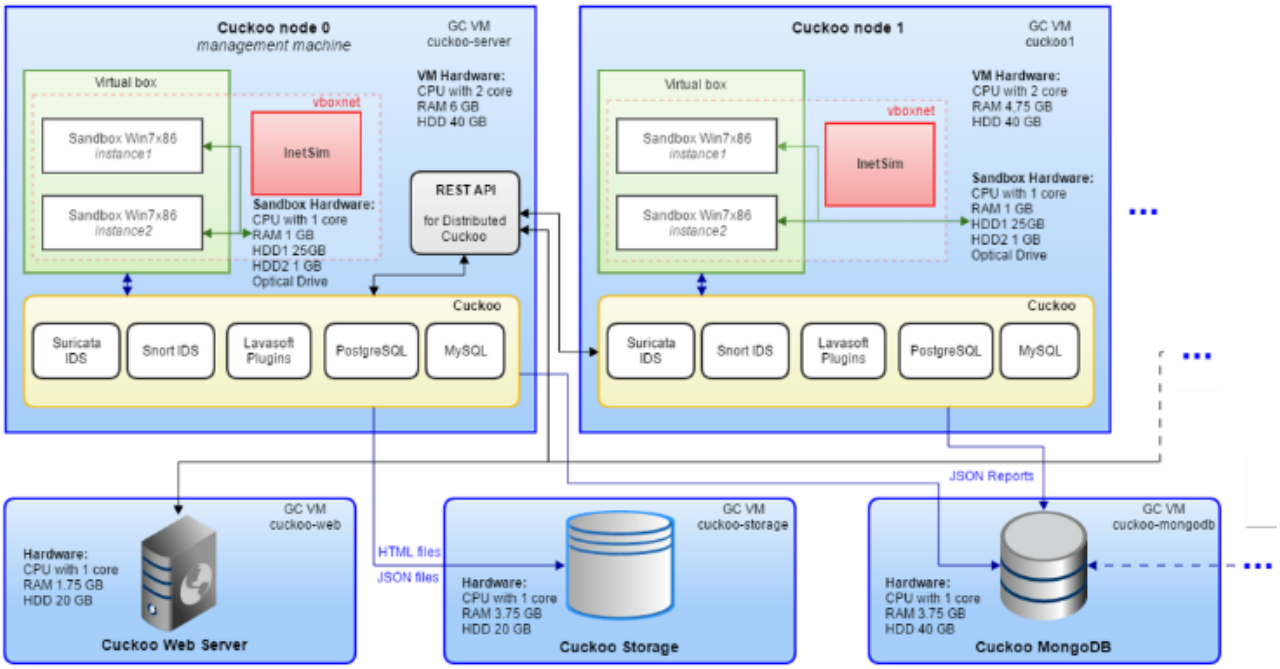


Fig. 8. **The main components of the distributed cloud service Cuckoo**
Source: **Compiled by the author**

CUCKOO MALWARE ANALYSIS CLOUD SERVICE

Cuckoo Sandbox is an open-source system for automatic investigation of malware, exploits, scripts, documents, archives, and links [8]. In a specially prepared virtual system, the Cuckoo agent is installed and added to the startup, which will be interacting with the sandbox. Network interfaces are specially configured to intercept and further analyze network traffic. After all the manipulations, a snapshot of the Snapshot file system is taken. The sandbox loads the file under test, determines its type, and performs the necessary manipulations per the file type. All changes inside the sandbox are saved in the report. Finally, the system restores Snapshot and the virtual environment returns to its original state (Fig. 8).

To expand the functionality of the Cuckoo system, additional plugins have been developed that significantly expand the functionality of the original solution to detect Zero-Day malware (Fig. 9).

The added plugins are taken from the MAS system and adapted to work in the Cuckoo environment. The plugins perform the following actions: search for anomalies in the Windows hosts file (Fig. 10), save memory dumps of malicious processes and code injected into the address space of processes (Fig. 11), monitoring of malicious rootkit activities.

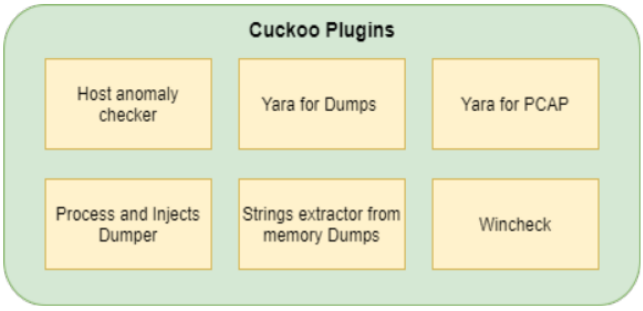


Fig. 9. **Additional plugins that extend the functionality of the Cuckoo system**
Source: **Compiled by the author**

To detect malicious activity the Yara rules scan is performed for memory dumps, network traffic, and malware executable file. Yara signatures for memory dumps allow to successfully detect malicious programs Rbot, Blazebot, Nrgbot [9, 10] (Fig. 12 and Fig.13), as well as their new polymorphic modifications, that are poorly detected by antivirus solutions.

At the time of the infection, only 10 % of the presented antivirus solutions detected a polymorphic sample. Such an approach allowed botnet developers to remain undetected for a long time, while the polymorphic modifications spread through the legal file hosting Dropbox and made it much more difficult for antivirus companies to block malicious URLs on the infected hosts.

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

```python
class Hosts(Auxiliary, Thread):
    """Find anomalies in Windows hosts file"""

    def __init__(self, options={}, analyzer=None):
        Thread.__init__(self)
        Auxiliary.__init__(self, options)

        self.hosts_path = os.environ['WINDIR']+ "\\System32\\drivers\\etc\\hosts"
        self.hosts_md5_original = 0
        self.do_run = True

    def checksum(self, item):
        try:
            fh  = open( item,"rb" )
            MD5 = hashlib.md5()
            while True:
                data = fh.read(8192)
                if not data:
                    break
                MD5.update(data)
            fh.close()

            return MD5.hexdigest()
        except Exception, e:
            log.exception ("[Hosts: checksum Exception] => %s", e)

    def stop(self):
        """Stop Hosts Analyzer"""
        try:
            hosts_md5_mod = self.checksum(self.hosts_path)
            log.debug("Hosts mod: %s", hosts_md5_mod)
            if str(self.hosts_md5_original) != str(hosts_md5_mod):
                upload_to_host(self.hosts_path, "hosts/hosts")
        except Exception, e:
            log.exception ("[Hosts: stop Exception] => %s", e)
        finally:
            self.do_run = False

    def run(self):
        """Run Hosts Analyzer"""
        try:
            self.hosts_md5_original = self.checksum(self.hosts_path)
            log.debug("Hosts original: %s", self.hosts_md5_original)
        except Exception, e:
            log.exception ("[Hosts: run Exception] => %s", e)
```

Fig. 10. **Cuckoo plugin for monitoring anomalies in the host file**

Source: **Compiled by the author**

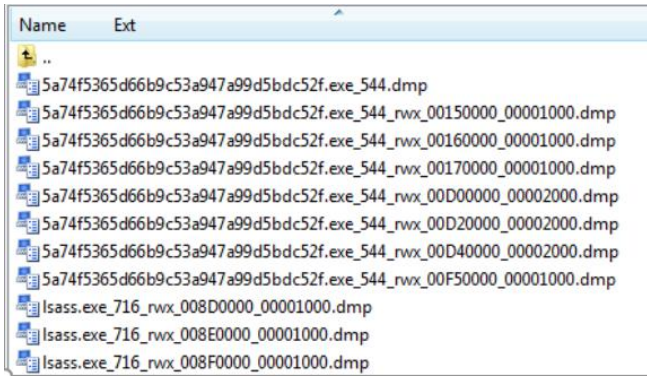


Fig. 11. **The dumps of a malicious file with MD5 5a74f5365d66b9c53a947a99d5bdc52f and dumps of malicious code injected into the address space of the lsass.exe process**

Source: **Compiled by the author**

The distributed version of the Cuckoo system does not contain a web interface by default and only works through the API. For the effective work of malware analysts, SOC engineers, and computer forensic experts, a web interface with the ability to send a file for analysis (Fig. 14) and receive scan results by e-mail was created (Fig. 15).

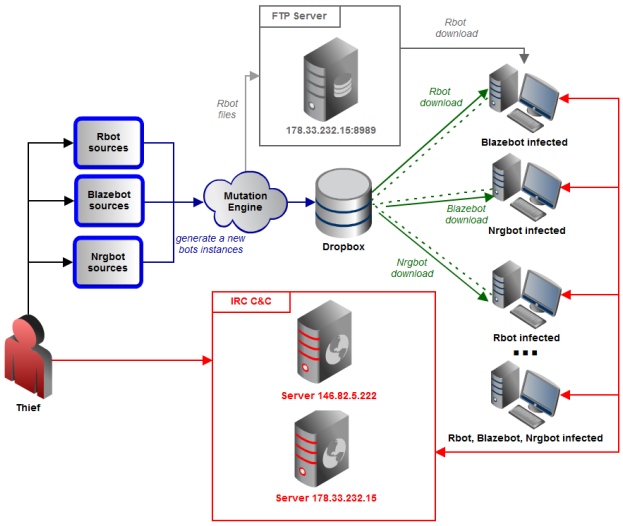


Fig. 12. **Scheme of the tandem of malicious programs Rbot, Blazebot, Nrgbot**

Source: **Compiled by the author**

The cloud-based malware analysis service automatically scales horizontally with a significant increasing the input stream of analyzed files. There is the possibility to create the new Cuckoo nodes in the public cloud if the input samples flow is highly increased and delete them in case of small amounts of input data. This approach makes it possible to save money that is spent on public cloud support. At the same time, the hybrid architecture allows to deploy nodes in the company's private cloud using its own data centers and not invest in public infrastructure if required.

METHODS

Methods for analyzing and diagnosing Zero-Day threats have been developed:

1) The automatic decision-making system Verdictor has been developed, which concludes the belonging of the sample to a malicious object based on the found anomalies in traffic fragments, memory dumps, and the method includes:

– preparation of samples of network traffic, strings extracted from the memory dumps of a malicious object;

– preparation of a unique set of strings analyzing CharsRate, TotalTrashRate, and the number of characters in a string;

– remove strings included in the Exclusion dictionary;

– creation of Yara rules;

– testing the rule based on MAS artifacts to search for polymorphic objects;

```
rule Rbot
{
    strings:
        $a1 = "ddos.random"
        $a2 = "ddos.ack"
        $a3 = "ddos.syn"
        $a4 = "SYSTEM\\CurrentControlSet\\Control\\Lsa"
        $a5 =  "Remote shell"
        $a6 = "TCP redirect"
        $a7 = "qwerty"
        $a8 = "databasepass"
        $a9 = "nokia"
        $a10 = "wwwadmin"

    condition:
        all of ($a*)
}
rule Blazebot
{
    strings:

        $a1 = "dcom135"
        $a2 = "dcom445"
        $a3 = "net start \"Terminal Services\""
        $a4  ="MSN// Message & Zipfile sent"
        $a5 = "Socks4 Server"
        $a6 = "Portscan: %s:%d open."

        $b1 = "B.l.a.z.e.b.o.t"

    condition:
        all of ($a*) or $b1
}
rule WormDorkbot
{
        strings:
                $a1 = "facebook" nocase
                $a2 = "twitter" nocase
                $a3 = "symantec" nocase
                $a4 = "threatexpert" nocase
                $a5 = "youporn" nocase
                $a6 = "vkontakte" nocase
                $a7 = "youtube" nocase
                $a8 = "admin" nocase
                $a9 = "letitbit" nocase
                $a10 = "lavasoft" nocase

        $b = "ngrBot"

        condition:
                ($a1 and $a2 and $a3 and $a4 and $a5 and
                $a6 and $a7 and $a8 and $a9 and $a10) or
                ($a1 and $a2 and $a3 and $a4 and $b) or
                ($a1 and $a2 and $a5 and $a6 and $a7 and
                $a8 and $a9)
}
```

Fig. 13. **Yara signatures for detecting polymorphic modifications**

Source: **Compiled by the author**

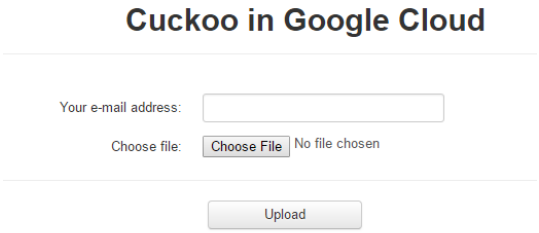


Fig. 14. **The web interface of the cloud service Cuckoo**

Source: **Compiled by the author**

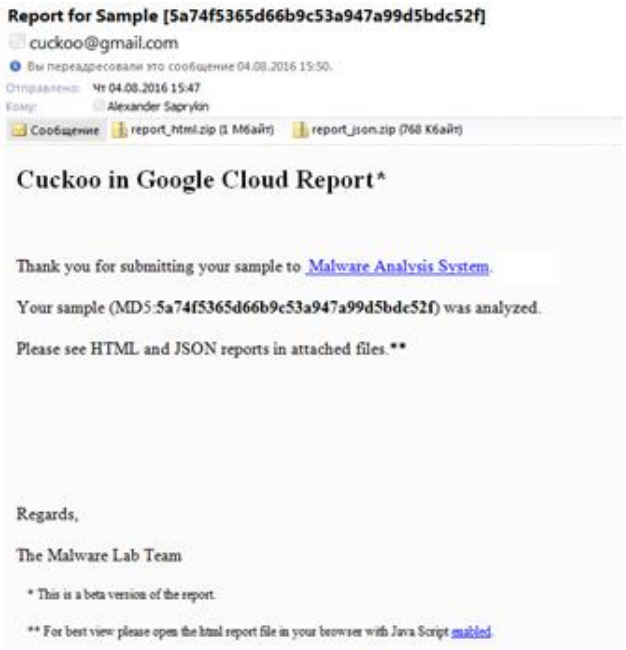


Fig. 15. **Example of a report sent by email from the Cuckoo system**

Source: **Compiled by the author**

All collected artifacts are stored in the MongoDB database after analyzing malicious samples (Fig. 16).

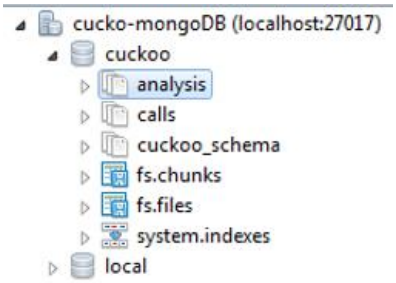


Fig. 16. **The architecture of the Cuckoo cloud service database**

Source: **Compiled by the author**

– recursive search for same strings found in the samples after the first iteration using difflib;

– Yara rule update;

– integration of the rule into the input set.

2) The system for static analysis of a malicious sample using a built-in multiscanner in the MAS system has been developed. The system allows the addition of Endpoint Protection antivirus programs for local file detection. The system includes the following components:

– the instance with the Windows operating system;

– preinstalled Endpoint antivirus solutions, maximum 3 per asset;

– python script for sending a file for analysis and receiving an antivirus verdict;

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

– saving results into av_detects_table MySql MAS artifact database;executable file using Yara rules (Fig. 17).

3) The core Sandbox module based on the Windows 7 operating system and VMware VIX API has been developed. The module allows analyzing malicious and non-malicious objects using a modular (plugin) architecture to expand functionality in the programming languages C ++, Perl, Python, Autoit. The module includes the following components:

Dumper – dumps of new processes in the system and memory dumps of injects in processes;

String from dumps – saves strings from memory dumps and injects;

URL extractor – parses Wireshark log and get all of the visited URLs;

ProcMon – is the plugin for catching file system and registry activities. Microsoft Sysinternals component;

WinCheck is a freeware tool that inspects undocumented or not enough documented Windows internal structures. This plugin catches anomalies in the system and we can detect new user-mode and kernel-mode rootkits even if we have no Yara rules for them;

Screenshoter – the module takes screenshots of malware windows;

File Handler – saves new dropped or downloaded files from malware and their paths;

Status Handler – creates a 0-size file with MD5 name in SandboxOutput folder to indicate the end of sandbox analysis:

\\ 192.168.50.163 \ sandboxoutput \ 2021-03-03 \ 0f041ac5d7d1acb3a9280743a909c330_done_15-35

Hosts file anomalies checker – collects anomalies from the Windows hosts file.

Mutexes checker is a component of Microsoft Sysinternals. The module collects information about the created malicious unique identifiers in the Windows system;

Launcher – executes the following types of files: EXE, DLL, URL, DOC, PPT, PDF.

RESULTS COMPARISION

The MAS malware analysis system allows to build your own virus laboratory to identify new malicious objects that are not detected by antivirus solutions, collects URL addresses from the files being examined, blocks traffic to the CnC servers via IDS / IPS solutions. The system allows to generate malware descriptions with step-by-step removal instructions [11]. The malware artifacts database allows to write own solution to automatically remove a new threat from cyberspace, while antivirus solutions do not have signatures.

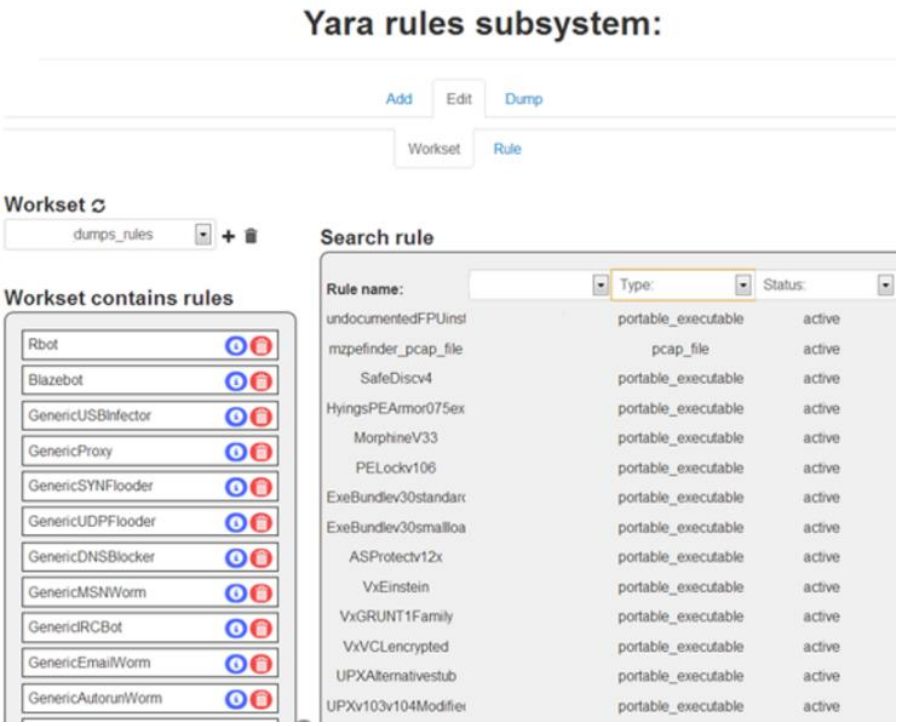


Fig. 17. **MAS module for managing Yara rules**
Source: **Compiled by the author**

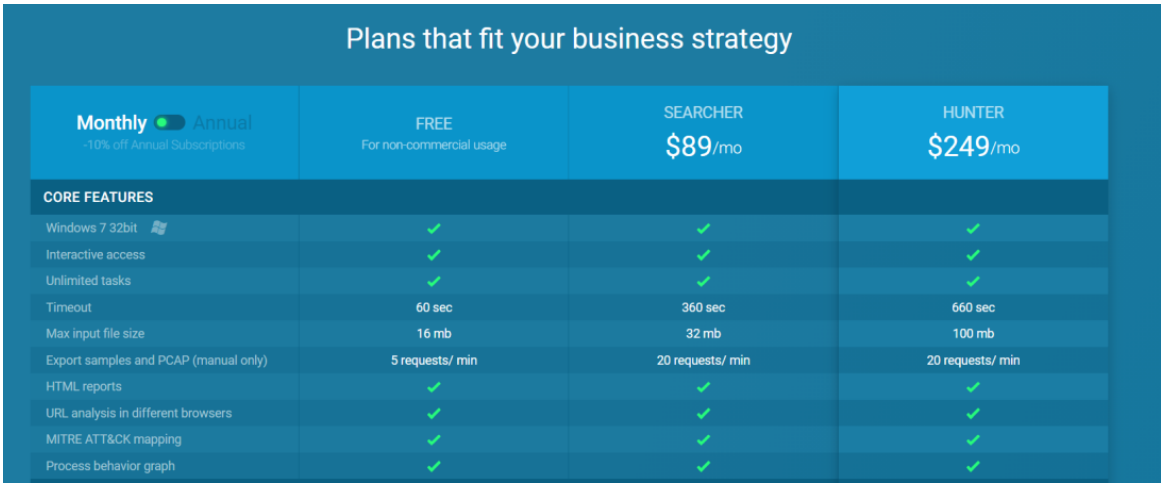


Fig. 18. **Malware analysis limits [15]**
Source: **https://app.any.run/plans**

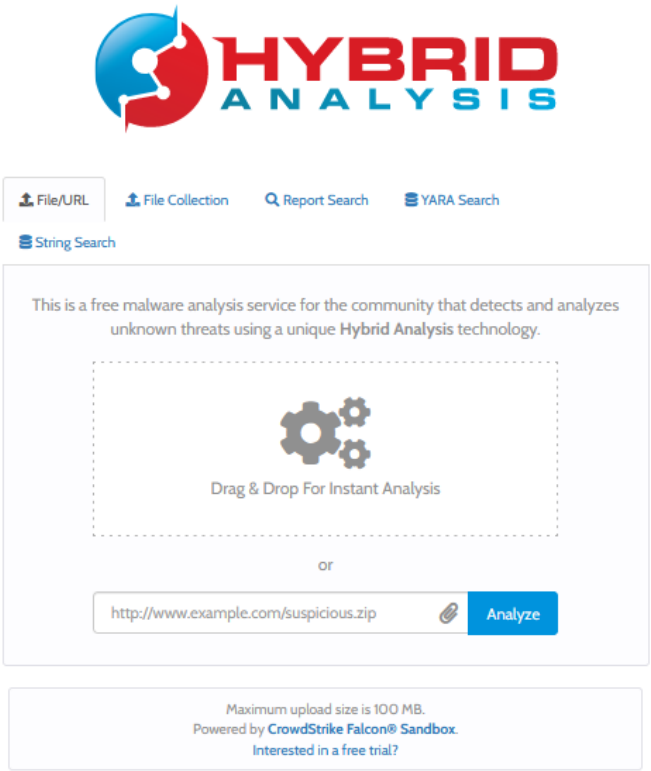


Fig. 19. **Malware analysis limits [16]**
Source: **https://www.hybrid-analysis.com/**

The advantage of the MAS system over the competitors [15, 16], [17, 18]:

1) MAS does not have an actual limit for malicious file analysis with 1GB or more in size. Sandbox instances can be easily scaled up and added the required number of resources. By default, the file size limit is 200 MB, but if necessary, these restrictions are removed. The file size limits for existing solutions are limited to 100 MB (Fig. 18 and Fig.19) even in paid subscriptions.

2) The malicious file analysis time in the MAS system is 4 minutes per one instance. To increase the processing speed of the malware flow the new Sandbox instances could be added. At the same time, even paid solutions have a limit of 20 requests per minute.

3) MAS contains a local multiscanner, while the existing analogs use only Virustotal results with the available limitation of 4 queries per minute. The main feature of the multiscanner is special system

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

settings that do not allow sending sensitive data to public space.

CONCLUSION

Scientific novelty and originality consist in the creation of the following methods:

1) detecting the sample by preinstalled antivirus solutions that allow static scanning in separate threads without requests restrictions for increasing the malware processing speed and restrict public access to confidential files;

2) diagnosing polymorphic malware using Yara rules, that allows detecting new modifications that are not detected by available solutions.

The proposed hybrid system architecture allows to perform a retrospective search by families, tracking changes in destructive components, collect the malicious URLs database to block traffic to C&C servers, collect dropped and downloaded files, analyze phishing emails attachments, integrate with SIEM, IDS, IPS, antiphishing and Honeypot systems, improve the quality of the SOC analyst, decrease the incidents response times and block new threats that are not detected by available antivirus solutions.The practical significance of the results is in the cloud service development that combines MAS Sandbox and a modified distributed Cuckoo sandbox, which allows to respond to Zero-Day threats quickly, store a knowledge base for artifacts correlation between polymorphic malware samples, actively search for new malware samples and integrate with cyber protection hardware and software systems that support the Cuckoo API. The proposed cloud service makes it possible to use neural network technologies for detecting malicious objects [12, 13], [14], [23], collect artifacts for creating Yara rules and detect polymorphic malware.

The multicomponent service that allows to organize a free malware analysis solution based on hybrid deployment architecture in public and private clouds has been developed. MAS can work with or without access to the Internet, showing high performance during malicious flow analysis in comparison with the considered solutions, detect high-level malicious complexes that are used polymorphic mutators, and Zero-Day vulnerabilities for their distribution.

REFERENCES

1. “WatchGuard’s Threat Lab Analyzes the Latest Malware and Internet Attacks”. Internet Security Report - Q3 2020. – Available from: https://www.watchguard.com/wgrd-resource-center/security-report-q3-2020. – [Accessed: Jan, 2021].

2. “Subscribe to the AV-TEST”. NewsletterWell-informed on Security. – Available from: https://www.av-test.org/en/statistics/malware/. – [Accessed: Jan, 2021].

3. “Global Sandboxing Market Size And Forecast To 2025”. – Available from: https: //www.verifiedmarketresearch.com/product/global-sandboxing-market-size-and-forecast-to-2025/. – [Accessed: Jan, 2021].

4. “Yara”. – Available from: http://virustotal.github.io/yara/. – [Accessed: Jan, 2021].

5. “Virustotal”. – Available from: https://www.virustotal.com/. – [Accessed: Jan, 2021].

6. “MySQL”. – Available from: https://github.com/mysql/. – [Accessed: Jan, 2021].

7. “Suricata”. – Available from: https://github.com/OISF/suricata/. – [Accessed: Jan, 2021].

8. “Cuckoo Sandbox repository”. – Available from: https://github.com/cuckoosandbox/cuckoo. – [Accessed: Jan, 2021].

9. “RBOT malware description”. – Available from: https://www.adaware.com/myadaware/malware-descriptions/blog/rbot. – [Accessed: Jan, 2021].

10. Adamov, A. & Saprykin, A. “Analysis and Detection of Polymorphic Spyware”. *Hakin9 Magazine. Software Press*. Warsow: 2013; Vol. 8 No. 01 Issue 01/2013 (61): 6–11.

11.” Malware encyclopedia”. – Available from: https://www.adaware.com/malware-encyclopedia. – [Accessed: Jan, 2021].

12. Saprykin, A., Kiktenko, V., Galagan, S. & Kunitsky, A. “Diagnosis Method of Malicious Code in Executable Files”. *Proceedings of the 5th East-West Design and Test Workshop*. Yerevan: Armenia. 7-10 Sept., 2007.

13. Saprykin, A. S. “Neural Network Methods Detection of Malicious Code in Software Objects”. *Eastern European Journal of Advanced Technology*. Kharkiv: Ukraine. 2009. No. 2/3 (38): 51–55.

14. Korablev, N. M. & Kushnarev, M. V. “Model of a Heuristic Analyzer of Malicious Programs Based on an Artificial Immune Network”. *Information Systems.* 2013. No. 8(115): 216–222.

15. “AnyRun Sandbox”. – Available from: https://app.any.run/docs/. – [Accessed Jan, 2021].

16. “Hybrid Analysis Sandbox”. – Available from: https://www.hybrid-analysis.com/. – [Accessed: Jan, 2021].

17. “Cape Sandbox”. – Available from: https://capesandbox.com/. – [Accessed: Jan, 2021].

18. “Cuckoo Sandbox”. – Available from: https://cuckoo.cert.ee/. – [Accessed: Jan, 2021].

19. Surkov, S. S. “Reduction of the Harmful Effect of Critical Modes in the Operation Queue Environment for Authorization Protocols for Large Requests”. *Applied Aspects of Information Technology. Publ. Nauka i Tekhnika.* Odessa: Ukraine. 2020; Vol. 3 No.3: 145–153. DOI: https://doi.org/10.15276/aait.03.2020.3.

20. Adamov, O. S., Hahanov, V. I., Chumachenko, S. V. & Abdullayev, V. G. “Blockchain Infrastructure to Protect Cybersystems”. *Radioelectronics & Informatics*. Kharkiv: Ukraine. 2018. No. 4 (83): 64–85. DOI: https://doi.org/10.30837/1563-0064.4(83).2018.184705.

21. Rucinski, Andrzej, Kovalev, I. S., Drozd, M. O., Drozd, O. V., Antoniuk, V. V. & Sulima, Yu. Yu. “Development of Computer System Components in Critical Applications: Problems, Their Origins and Solutions”. *Herald of Advanced Information Technology. Publ. Nauka i Tekhnika.* Odessa: Ukraine. 2020. Vol. 3 No. 4: 252–262. DOI: https://doi.org/10.15276/hait.04.2020.4.

22. Surkov, S. S. & Martynyuk, O. M. “Improvement of Security for Web Services by Research and Development of OAuth Server”. *Electrotechnic and Computer Systems*. Odesa: Ukraine. 2016; Vol. 23(99): 99–105. DOI: https://doi.org/10.15276/eltecs.23.99.2016.16.

23. Semenov, S. H., Havrylenko, S. Yu., Hloba, S. M. & Babenko, O. S. “Development of Computer Viruses Detection System Based on ART-1 Neural Network”. *Information Processing Systems*, 2015; Vol. 10(135): 126–129.

24. Adamov, O. S. & Hahanov, V. I. Signature-Qubit Methods Recognition Destructive Codes (in Ukrainian). *Radioelectronics & Informatics*. Kharkiv: Ukraine. 2019. No.1 (84): 35–53. DOI: https://doi.org/10.30837/1563-0064.1(84).2019.184719.

25. Hahanov, Vladimir. “Cyber Physical Computing for IoT-driven Services”. New York: USA. *Publ. Springer*. 2018. 279 p. DOI: https://doi.org/10.1007/978-3-319-54825-8.

26. Carlsson, A. & Adamov, A. “A Sandboxing Method to Protect Cloud Cyberspace”. *IEEE East-West Design & Test Symposium (EWDTS)*. 2015. p. 1–3. DOI: https://doi.org/10.1109/EWDTS.2015.7493177.

27. Kolbitsch, C., Comparetti, P. M., Kruegel, C., Kirda, E., Zhou X.-Y. & Wang, X. “Effective and Efficient Malware Detection at the end Host”. Proc. USENIX Secur. Symp. Aug. 2009; Vol. 4 No. 1: 351–366.

28. Ding, Y., Xia, X., Chen, S. & Li, Y. “A Malware Detection Method Based on Family Behavior Graph”. Comput. Secur. Mar. 2018; Vol.73: 73–86. DOI: https://doi.org/10.1016/j.cose.2017.10.007.

Conflicts of Interest: The authors declare no conflict of interest

Received 09.12.2020
Received after revision 11.05.2021
Accepted 15.06.2021
DOI: https://doi.org/10.15276/hait.02.2021.5
УДК 658 + 512.011 + 681.326 + 519.713

Моделі і методи діагностування Zero-Day загроз в кіберпросторі

Олександр Сергійович Саприкін
ORCID: https://orcid.org/0000-0001-5399-5054; sashasapr@gmail.com
Харківський національний університет радіоелектроніки, пр. Науки, 14. Харків, 611166, Україна

АНОТАЦІЯ

Робота присвячена розробці моделей і методів діагностування Zero-Day загроз в кіберпросторі для підвищення ефективності виявлення складних шкідливих комплексів, що використовують поліморфні мутатори. Пропонується метод детектування досліджуваного зразка антивірусними рішеннями за допомогою публічного і локального мультисканера. Розробляється метод діагностування поліморфних шкідливих програм за допомогою Yara правил. Описується багатокомпонентний сервіс, що дозволяє організувати безкоштовне рішення аналізу шкідливих програм з гібридною архітектурою розгортання в публічних і приватних хмарах. Виконується проектування хмарного сервісу для детектування шкідливих програм на основі пісочниць з відкритим вихідним кодом і MAS, що дозволяє горизонтально масштабироваться в гібридних хмарах і показує високу пропускну здатність при обробці потоку шкідливих і невредоносних об'єктів. Основним завданням

сервісу є збір артефактів після динамічного і статичного аналізу досліджуваного об'єкта для детектування Zero-Day загроз. Показується ефективність запропонованих рішень. Наукова новизна дослідження визначається створенням методів:
1) детектування досліджуваного зразка заздалегідь встановленими антивірусними рішеннями, які дозволяють в окремому потоці проводити статичне сканування досліджуваного об'єкта без обмежень на кількість запитів за хвилину і тим самим підвищити швидкість обробки об'єктів і обмежити публічний доступ до конфіденційних файлів; 2) діагностування поліморфних шкідливих програм за допомогою Yara правил, що дозволяє детектувати нові модифікації, які не виявляються доступними рішеннями. Практична значущість визначається розробкою гібридної архітектури системи, яка дозволяє проводити ретроспективний пошук за сімействами, відстежуючи зміни в деструктивних компонентах, збирати базу шкідливих URL адрес для блокування трафіку до керуючих серверів, збирати витягнуті завантажені файли, аналізувати вкладення в фішингові листи, інтегруватися з SIEM, IDS, IPS, антифішинг і Honeypot система, поліпшити якість роботи SOC аналітика і час реакції на інциденти, упереджувати атаки зловмисників і блокувати нові загрози, що не детектируются доступними антивірусними рішеннями.

Ключові слова: Zero-Day загрози; Sandbox; хмарний сервіс; антивірус; діагностування; детектування; поліморфні коди; сканер; емулятор

ABOUT THE AUTHORS



Oleksandr S. Saprykin – Postgraduate Student. Kharkiv National University of Radio Electronics, 14, Nauky Ave. Kharkiv, 61166, Ukraine
ORCID: https://orcid.org/0000-0001-5399-5054; sashasapr@gmail.com
***Research field*:** Security Engineering; System Engineering; Automation Malware Analysis Systems; Machine Learning; Neural Networks

Олександр Сергійович Саприкін – аспірант. Харківський національний університет радіоелектроніки, пр. Науки, 14. Харків, 611166,Україна