UDC 004.056.5

MODERN INFORMATION SYSTEMS SECURITY MEANS

A. I. Verner, I. A. Klymenko

The article provides a thorough review of current research on information security means available for the endpoint protection. In the first chapter, categories and features of types of threats to information security are considered. The second chapter provides a general description of threat analysis methods, compares static, dynamic, hybrid malware analysis methods and highlights the advantages and disadvantages of each of them. The third chapter considers the modern methods of detecting and mitigating threats to information systems, as well as the peculiarities of their implementation. The purpose of this article is to provide a general overview of the current state of information security and existing modern methods of protecting information systems from possible threats.

Key words: *information security, information systems, security means, malware.*

Introduction

High rates of technological progress and information technologies dissemination are ubiquitous nowadays. Statistical data [23] indicate that there is a concurrent pattern of yearly exponential growth in the volume of harmful software affecting information systems. This necessitates the creation of a variety of novel, adaptable forms of protection mean. Nevertheless, despite the coordinated efforts of experts, the problem of malware analysis and detection is still unresolved.

As of September 2022, research on operating system (OS) use [19] reveals that the commercial Windows OS continues to be the most popular among desktop computer users. Regarding the threats, despite the Q4 2021 Internet Security Report's conclusions that attacks were decreasing downward year over year, a large increase in threats detections in Q1 2022 indicated that the situation became worse [11]. In Q2 2022, 55,314,176 malicious and potentially unwanted objects were detected by security systems [13].

Additionally, the predominance of embedded systems, which are mostly employed in the so-called "Internet of Things" (IoT), is growing quickly. This has caused some obvious shifts in the landscape of malicious software. Due to the high rates of product release, corporations pay insufficient attention to the issue of product security, which leads to the presence of a significant number of major vulnerabilities in such systems being rather often detected. Architecturally embedded systems have strong differences from desktop personal computers, which is caused, first of all, by the use of various processors and rather limited resources. From the point of view of the operating systems usage here, of course, the situation is also radically different, since according to [22] developers use Unix-like OSes with various variations of the Linux kernel. According to evidence [12], almost half of smart homes with built-in systems had critical vulnerabilities that allowed attackers to easily attack them. The report [13] shows, that most of the IoT devices were attacked using the Telnet protocol, as before (Telnet – 82,93%, SSH – 17,07%). In addition, according to this source, there is a 217% increase in the number of attacks compared to previous years.

By analyzing the aforementioned facts, we can draw the insight that creating protection means to increase the security of information systems is a very critical issue in the contemporary.

This paper describes the actual status of information security, categorizes and highlights the specific attributes of security mechanisms depending on attacks, and examines contemporary methods for detecting threats to information systems.

Information security threats: categories and specifics

A threat in the context of information security is a potential negative action or occurrence facilitated by a vulnerability, leading to an unintended effect on a computer system or application.

Threats to information security can take a variety of forms including software attacks, intellectual property theft, identity theft, equipment theft, information theft, sabotage, and information extortion. Any software that has the potential to compromise the integrity of an information system

is considered malware [17]. Malware is an acronym for malicious software and, therefore, it is essentially defined as harmful software that can be invasive computer code or anything else created with the intention of harming a system. Due to the presence of many malicious software and a huge range of programs, each type of malware can be unambiguously divided into classes. Most time it is incorrectly interpreted that, viruses, worms, and bots are all the same things. The only common feature is that they are all related to the malicious programs, however they behave in the most distinct way. As was mentioned, malware includes viruses, worms, Trojan horses, rootkits, spyware, keyloggers, etc. According to the report [1], most spread were the heuristic malware in 2021. The fig. 1 represents the graph of detected malicious software for the 2021 year.

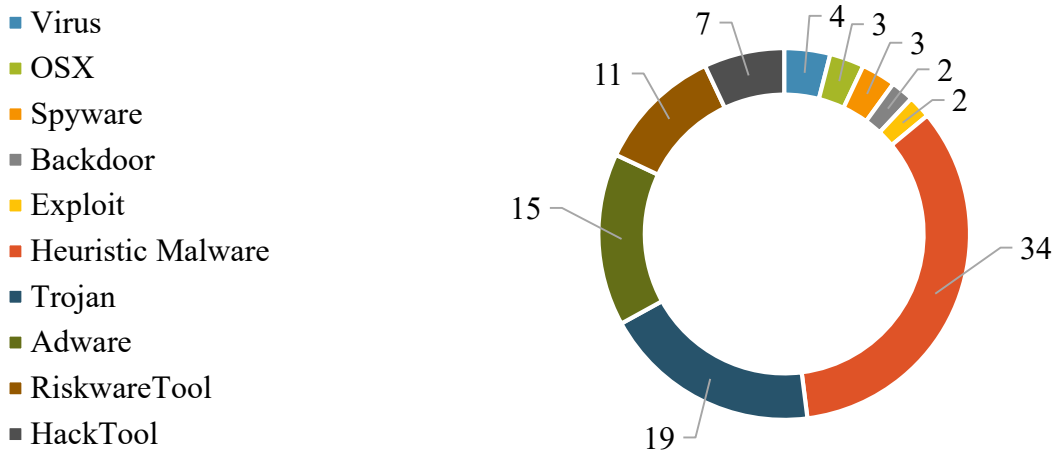

Fig. 1. Top 10 threats detection categories 2021.

The intricacies of how each form of malicious software functions will be later covered in more depth further.

Generally, the malware may be divided into two main groups by such aspects:

- Methods of Infection
- Malware Behavior

Based on the infection methods the following examples of malware can be identified:

The ***viruses*** are malicious software having a self-replication mechanism. Such software has an executable file that it uses to replicate itself to other host systems and proliferate. Because the program is passive, infection happens through files, media, or network files. The viruses could also modify its replicated copies, depending on how complicated the computer code is [21]. Viruses can be used to steal information, build botnets, show adverts, and many other activities in addition to harming computer nodes and networks.

Although ***worms*** also have a replication mechanism, they are representing an active malware that spreads over the network by taking advantage of numerous vulnerabilities in the existing software or operating system. They include malicious processes in them that may be utilized to create channels of communication and operate as active carriers. This class drastically lowers the system's performance and continuously scans its resources [20], which causes the node to become unstable and, in severe circumstances, the system to crash. Moreover, worms could produce payloads in the form of several bits of code that are created to harm the node by stealing data, erasing files, or building a bot that can connect an infected device to a botnet [21]. Unlike viruses, worms do not require human activity to spread, as they could spread and reproduce independently.

A ***Trojan*** is a piece of software that has the appearance of being trustworthy but when downloaded and run, runs any contained harmful code or files. A Trojan may have no payload or have extra malware installed in the form of viruses. Trojans, in contrast to viruses and worms, do not have a mechanism for self-replication and are only triggered when users launch them. However, the payload can include malware that enables an attacker to remotely access the computer node and carry

out any nefarious deeds. The effects of Trojans programs on PCs vary depending on the extra payload and are typically enhanced by social engineering [9, 15].

A ***backdoor*** is a hidden "entrance" used to gaining access. It is occasionally made specifically by service providers as a remote tool for system checks, troubleshooting, and diagnostics. The simple existence of a backdoor is a huge security risk, as it is not difficult to detect. Attacks frequently occur as a consequence of safe backdoors, for instance with the "backdoor" virus. This type of malicious software allows for remote, illegal access to a computer system or application by taking advantage of system weaknesses and shortcomings. It operates in the background, much like any malicious software. This access provides the full range of actions to perform malicious operations on the system. Computer nodes are very vulnerable to illegal copying of files, modifications, data theft by using backdoors [26].

The ***bots*** are computer programs created to carry out particular tasks. Bots were initially created to control chat channels. While some of them are exploited for legal purposes, malicious bots are built to create botnets. A botnet is a network of node computers (zombies/bots) controlled by an attacker or botmaster. Bots infect and control another computer, which in turn infects other connected computers, forming a network of compromised computers botnet. Bots are frequently employed as spammers, for DDOS attacks, web distributors for spreading malware on file sharing, etc. The CAPTCHA tests are one tool used to defend systems against bots [9].

The behavior-based malware can be also divided in multiple parts.

A ***spyware*** is malicious software that monitors user activities by accessing operating system features. Such spyware occasionally contains extra capabilities, including the ability to impede network connections or even modify the infected system's security settings. Spread occurs by attaching to legitimate software, Trojan horses, or even through known vulnerabilities. Spyware can monitor user behavior, for example, by collecting keystrokes and sending information to a remote host to an attacker [27].

Spyware includes ***keyloggers***. Such software performs the recording in the background. The user is unaware that a recording of the keys they press on the keyboard exists. The collected data is then transmitted to the attacker over the Internet. These applications are designed to steal passwords, such as those used for online banking. They can also employ spyware to steal other types of personal data, such digital documents. Spyware and keyloggers may be downloaded to a distant node via a variety of methods. The most common is by following a link in spam e-mails or by visiting web pages designed solely to infect nodes. Even still, this kind of malware is occasionally referred to as "Trojan," as it spreads similarly to Trojans.

The ***zombies***. They function in a manner akin to spyware. The infection method is the same, except instead of spying and stealing data, they wait for a hacker's order.

An ***adware***, performs automatic display of advertisements in the form of pop-up ads on websites, etc. Most of this software is designed to assist marketers produce products that will make companies money. Some adware packages contain spyware, which might eventually have serious repercussions including tracking user activities and information theft [24].

The ***rootkits*** are the advanced and complex applications typically developed as tools to conceal regular operations on the infected node. To prevent being discovered by the system, rootkits employ a number of tools. They are extremely intrusive and challenging to get rid of because they are undetectable. They are developed with the possibility of full control over the system and obtaining the highest privileges on the infected node [9]. The majority of node protection software solutions are ineffective in identifying and removing rootkits because to their use of cloaking methods. Monitoring the computer system's activity in respect to the topic of unexpected activities, analyzing memory dumps, and scanning system file signatures are other ways to counteract them.

A ***ransomware*** infects computing nodes or a network and keeps the system locked down, demanding a ransom from users. To prevent users from accessing the infected machine, the files are often encrypted or the system is banned. Then messages appear demanding payment in order to view the data. Such malware uses the same spreading techniques as computer worms.

Although there are additional varieties of malware, these are the most well-known and widespread today. A trend towards fewer new harmful software creations and a dramatic increase in

the overall quantity of malicious samples can be observed by examining the report of the past 10 years [25]. The graphs in Figure 2 and Figure 3 provide the visual representation of the problem.

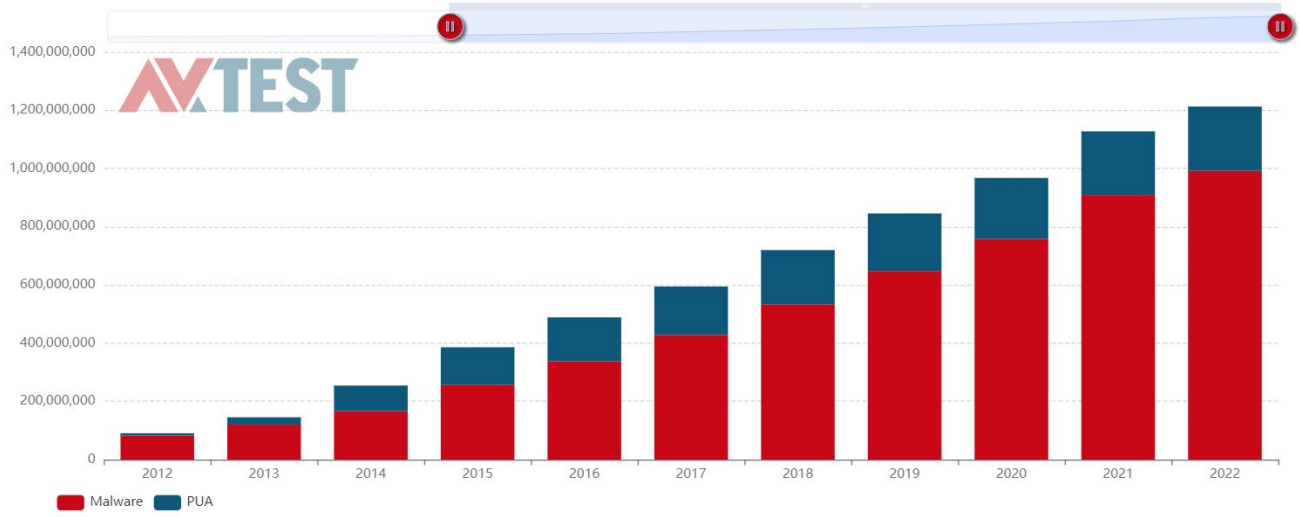

Figure 2. Total amount of malware and potentially unwanted applications (PUA) [25].

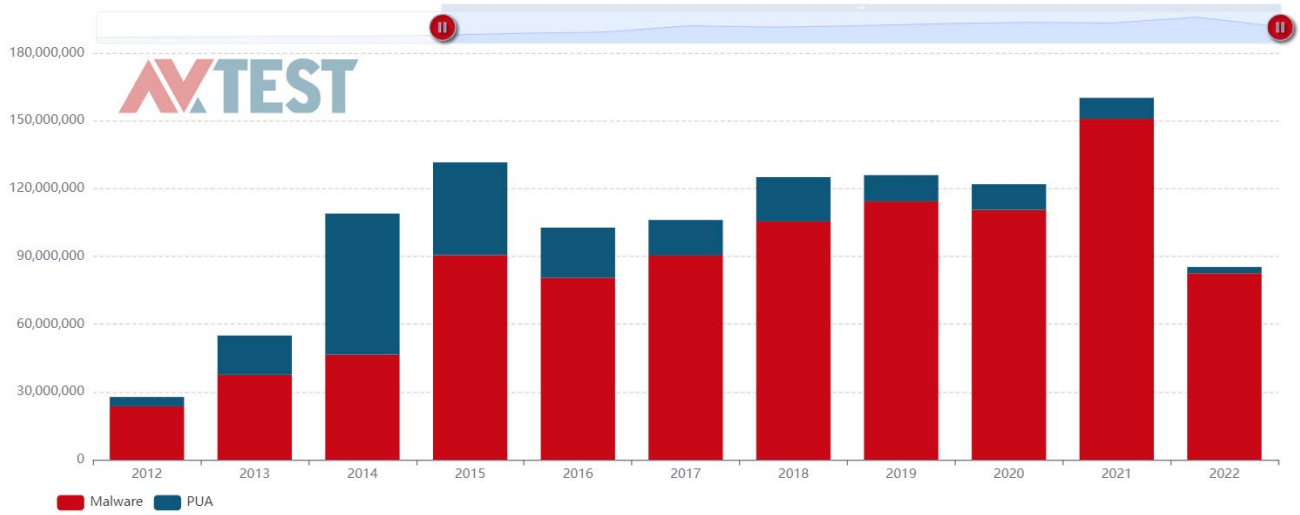

Figure 3. The annual increase of malware and PUA [25].

From the overall evolution of new malware over the past ten years, malware prevalence is rising yearly. At the time this article was published, there were more than 1.2 billion harmful programs in use in 2022 (according to the info from av-test.org report).

The analysis and detection technologies are continually improving as a result of the fight against malware samples. Malware detection technology has been continuously evolving from rule matching and feature code extraction in the early phases to dynamic and static detection and heuristic detection in the middle phases, and finally to the current machine learning and multi-engine joint learning. Nevertheless, anti-detection technology is also improving to overcome different anti-killing methods, malware employs shell, obfuscation, virtual machine protection, and other technologies [10]. In the next chapter we will consider the modern threats detection means.

Threat analysis techniques

Malicious threats can be detected using a variety of code analysis techniques. In general, such analysis methods can be separated into three main categories: hybrid, dynamic, and static. These techniques identify and classify malicious software and take action against it in order to protect computer systems from a potential loss of data and resources.

One of the first historically developed method is the ***static analysis***. The term "static analysis" describes the examination of harmful software without actually running it. String signatures, byte sequences, n-grams, library syntax calls, control flow graphs, opcode frequency distribution analysis, and other detection patterns are employed during static analysis. The analysis is carried out by preliminary file's unpacking and decoding of the execution file. Debugging and memory dump analysis tools are used to reverse-engineer the basic principles of how malicious software functions. Disassemblers and debuggers allow displaying the malware code in the form of assembly instructions, which provides information about what the malware actually does, and helps identify patterns to identify attackers. Such technique is very useful for analyzing the packaged executables that are challenging to disassemble.

However, such analysis loses its effectiveness in case the obfuscation is performed. The binary obfuscation techniques convert malware binaries into self-compressed and distinctively organized files. This is generally done to prevent modification and complexifying of the overall exploration of harmful software, thus additionally reducing the opportunity of obtaining any qualitative findings. Additionally, as mentioned in [6], when binary executables are used (obtained by compiling the source code) for static analysis, details like the size of data structures or variables are lost.

Technical methods used by attackers to evade static analysis led to the development of ***dynamic analysis***. The drawbacks of the static analysis methodology were studied by Moser [19]. The scientist developed a coding obfuscation-based method that shows static analysis is inadequate for identifying or categorizing malicious software. According to the conducted studies it is confirmed that since dynamic analysis is less vulnerable to obfuscation than static analysis, it serves as an essential supplement to static analysis.

Dynamic analysis of malicious programs includes the analysis of the program during its operation in the system [16]. The malware is executed in a secure and controlled environment, to avoid the transfer of the investigated malware to other systems or networks. Observation, samples gathering and the samples interactions with the system is the foundation of dynamic analysis. For this, the snapshot of the initial state of the virtual machine is taken before the malware is launched to execute on the test system. To examine changes, the input and output states are compared. After the changes obtained from observations, they are used to further remove malware from infected nodes and/or to simulate effective signatures. Like basic static malware analysis, dynamic analysis is an important initial step in malware analysis, although it does not provide comprehensive information about the malware [8].

Extended dynamic analysis involves the use of tools to study the state of the malicious program during its execution. For instance, this allows to study the harmful code's internal state. The use of advanced analysis techniques provides information that cannot be collected using other methods [20]. Dynamic analysis is always carried out in an isolated setting to ensure that all system inputs and outputs are known for further analysis. The use of additional tools also allows to perform tracking of the APIs used at this stage, to check the system functions calls, called and deleted files, registry changes, and data processed by the program analyzed during interaction with the system. Analyzing the parameters used in API and function calls allows semantic grouping of the functions used while, analyzing the processed and distributed data in the system provides insight into the files used and produced by the malware. This allows to determine the purpose of the malicious software development [2]. The advanced dynamic malware analysis is very useful for detecting malware variants and obscured techniques. Automated dynamic malware analysis tools are employed for convenience, and they produce reports that may be utilized in order to classify harmful samples based on their behavior.

By combining both static and dynamic analysis techniques the new threat analysis approach was developed – the hybrid analysis. Such a method benefits from both approaches. A software is first examined by code analysis and malware signature validation, following which it is launched in a virtual environment to ascertain its true behavior. This allows investigating the malicious software deeply.

It is important to identify the unique peculiarities of how each type of analysis is performed:

- the static analyzers, process executables without running them and extract the classification-related information from the binaries and their metadata;
- the dynamic analysis systems execute binaries in a virtualized environment and record sample behavior, isolating the indicators of malicious activity;
- the hybrid analyzers can analyze the encrypted malware being more precise and time consuming.

While all the approaches have positives and negatives, many endpoint security solutions tend to be handled by static analyzers because of the strict time constraints required to avoid impacting system performance.

The pros and cons of using each analysis technique are briefly illustrated in the table of general approaches comparisons (Table 1).

Table 1.
Threat analysis approaches comparisons [15]

Analysis approach	Static	Dynamic	Hybrid
Pros	– Efficient. – Low influence on performance. – Safer as does not require software execution. – High accuracy.	– Has better accuracy over static analysis.	– Far superior to static and dynamic analysis. – Can detect malware that is both known and undiscovered. – Can analyze the encrypted malware
Cons	– Unknown and encrypted malware cannot be analyzed. – Unable to recognize obscure malware.	– Unsafe and time consuming – High resources utilization	– Most time and resources consuming. – Most complex

Malicious threat detection techniques

Numerous techniques for identifying threats are developed as academic study on malware detection increases. Let's examine the primary methods for detecting malicious software on computing nodes in more detail.

It is impossible to categorically say that one method is superior to another when it comes to finding significant traits because each strategy is unique and has its own benefits as well as disadvantages.

Using behavioral modeling, heuristics, and simulation-based threat detection approaches, a large amount of malware can be identified. In addition, these models also allow detecting a new species of malware. However, they are not universal and cannot detect all the malicious software developed. Therefore, there is a need to find a method that would effectively detect even more complex, still unknown programs. Overview of malware detection approaches, features, and used techniques can be seen in Figure 4.

The ***signature-based*** detection technique was initially common. A signature is a feature of the malware that encapsulates the structure of the program and identifies each malware as unique. This technique rapidly and effectively recognizes known malware species. That is why the signature detection approach widely used in commercial antivirus applications.

This approach is fast and effective enough to detect known types of malware, but not powerful enough to detect unknown types of malware. Therefore, malicious software that exploits the zero-day

vulnerabilities cannot be identified by using such methodology. Additionally, by utilizing obfuscation, malware from the same species can easily avoid detection by signature-based methods [10]. As the method has such weaknesses, later other techniques emerged.

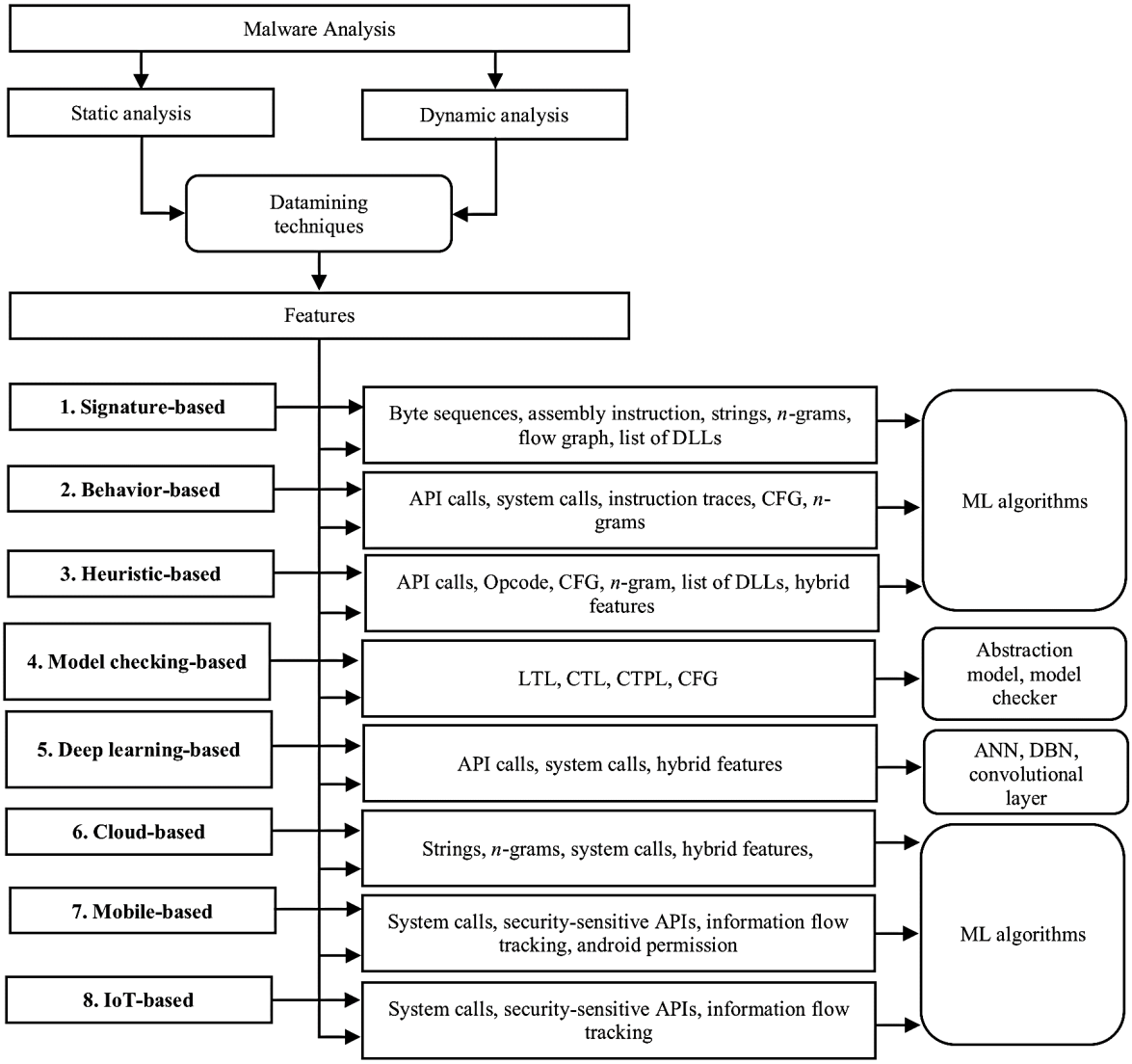

Figure 4. A diagram illustrating malware detection techniques and tools [4].

A ***behavioral method*** to malware detection uses monitoring tools to track the activity of the program and assess whether it is malicious. This method may be used to recognize the majority of new harmful software since behavior does not really change even when the software code does [5].

A malware sample could be incorrectly classified as harmless since some malware programs do not work properly in a secure environment. In behavior-based detection, features are first excluded from the dataset using data mining, and behaviors are then identified using one of the methods mentioned above. Then, using ML algorithms, particular characteristics from the dataset are retrieved and classification is performed.

A ***heuristic technique*** to malware detection has been popular in recent years. It is a sophisticated detection technique that draws on knowledge and a variety of approaches, including rules and machine learning (ML) techniques [3]. This method offers the opportunity to identify zero-day vulnerabilities, however it is unable to detect sophisticated software.

A ***model checking*** based approach. In this approach, malware behaviors are defined manually, and groups of behaviors are coded using linear temporal logic (LTL) to represent relevant features. Programmatic behavior is created by looking at the flow relationships of one or more system calls and defining the behavior using properties such as hiding, propagating, and injecting [3].

By comparing these behaviors, it is determined whether the program is malicious. This technique enables the detection of certain new software, but it cannot be used to detect a new generation of dangerous software.

Deep learning is a type of ML machine learning that inherits from artificial neural networks (ANNs) that learn from examples. This is a new approach that is widely used for image processing, drone control and voice control; however, it is still underutilized for malware detection. Although it is quite effective, its main drawback is that it is not resistant to attacks that use evasion.

Cloud computing is quickly expanding because it offers several benefits such as simple access, on-demand storage, and cost savings. Because the cloud is so widespread, it has also been used to identify viruses. With significantly larger malware databases and heavy computing resources, cloud-based malware detection improves detection performance for PCs and mobile devices.

Cloud-based detection employs several sorts of detection agents on cloud servers and provides security as a service. A user may submit any sort of file and obtain a report indicating whether the file is malware (e.g., Virus Total platform). Despite its benefits, this detection architecture has certain drawbacks.

Some drawbacks include the following:

- The cloud detection mechanism has some overhead over other detection mechanisms, so communication between the client and server must be optimized, especially for the Internet of Things and mobile devices.
- User must upload content to the cloud, which may reveal some sensitive data, such as location, password, and credit card information.
- The absence of real-time monitoring across all resources for all files.

The Internet of Things (IoT) architecture is composed of a wide range of Internet-connected smart devices such as household appliances, network cameras, and sensors. IoT and mobile devices have begun to outnumber PCs on the Internet. As mobile and IoT devices become more popular among consumers, they also become increasingly popular targets for attackers. As a result, the malware detection paradigm landscape is shifting away from desktops and toward IoT and mobile devices.

A novel approach for identifying DDoS malware in IoT contexts proposes malware categorization using convolutional neural networks and malware binary image analysis [14]. Being fast and lightweight, the mentioned method remains vulnerable to complex code obfuscation techniques. Partially this can be fixed by using static sequences and calls features limited to a certain degree.

One more method describes the detection of the cryptoransomware in IoT networks based on energy consumption footprint [7]. To accomplish malware application categorization, this technique likewise employs ML algorithms and tracks the energy consumption trends of several activities. However, the technique description proposed is unclear. Furthermore, there is no information on which ransomware family was examined or how they dealt with unknown malware.

Finally, lets briefly outline the benefits and drawbacks of each of the methods discussed above.

The signature-based approach allows performing the fast and efficient detection of known software. This method also proves its efficiency in malware detection in case the samples belong to same species. Unfortunately, such threat mean is unable to detect new types of malwares or the modification of the old one. Furthermore, it is not resistant to obfuscation and polymorphism.

The behavior-based approach has proven its validity for identifying the new malware types as it determines the malware functionality. Such method also allows detecting different species of the same malware being effective against polymorphism and obfuscation. One of the mentioned method's drawbacks is that it may produce the false positives due to the difficulty of the malicious and normal behavior separation.

Unlike above-mentioned approaches, the heuristic-based method allows detecting the unknown malware by using the combination of the static and dynamic analysis features. However, this way is a bit complex as it contains various number of rules and training phases being vulnerable to metamorphic techniques.

The model checking-based approach is complex and resource-intensive technique. However, it allows detecting the malware from the same family and is resistant to the polymorphism and obfuscation techniques.

One of the most powerful and effective is the deep learning-based approach. This method consumes some time during the detection and is not resistant to evasion attacks.

To enhance the detection performance for PCs the Cloud-based solution can be used. It provides better computational resources and bigger malware databases. Additionally, it can be easily accessed, managed and updated. However, as cloud is the remote source, some sensitive data leaks are also possible. Additionally, it requires continuous connection between the client and the server.

The last approach becomes more common nowadays due to the wide spread of the IoT devices. This approach similarly to the previous allows using both the static and dynamic analysis feature being limited to the uncomplex malware only.

Conclusions

Although the new approaches for the security means are being developed and enhanced daily, there is a still strong need in the development of the threat detection methods due to the prevalence of the malicious software nowadays. The article provided a thorough review of current research for malware detection methodologies, as well as techniques and algorithms utilized for malware detection. The benefits and drawbacks of each malware detection method have been discussed.

The most significant disadvantage of current security measures is their sensitivity to obfuscation. The use of deep learning methods as the foundation of the developed technique will allow eliminating the major vulnerability of the most often used security methods – identification of unknown forms of malicious software.

It is also shown that the percentage of new threat semantics decreases as a result of the fact that new instances of malicious software are only modifications of already implemented threat mechanisms to which polymorphism and obfuscation have been applied in order to change their signatures. Such a trend is positive, as it allows to significantly increase the security of information systems by preventing the execution of a considerable amount of malicious software in case of the specific approach development which will allow detecting and preventing threats resistant to such modifications.

References

1. 2022 Threat review Report [Electronic resource] // Malwarebytes Inc.. – 2022. – C. 32. – Resource access mode: https://www.malwarebytes.com/resources/malwarebytes-threat-review-2022/mwb_threatreview_2022_ss_v1.pdf
2. A survey on automated dynamic malware-analysis techniques and tools / M.Egele, T. Scholte, E. Kirda, C. Kruegel., 2012. – 6 P.
3. Alzarooni K. Malware variant detection : дис. канд. техн. наук : Dept. Comput. S / Alzarooni – London, 2012.
4. Aslan Ö. A Comprehensive Review on Malware Detection Approaches / Ö. Aslan, R. Samet. // IEEE Access,. – 2020. – P. 6249–6271, doi: 10.1109/ACCESS.2019.2963724.
5. Aslan Ö. Investigation of possibilities to detect malware using existing tools / Ö. Aslan, R. Samet. // IEEE/ACS 14th Int. Conf. Comput. Syst. Appl. (AICCSA). – 2017.
6. Chapaneri R. Staticand dynamic malware analysis using machine learning / R. Chapaneri, S. Shivshankar, C. Raghuraman. // Springer. – 2020. – P. 793–806.
7. Detecting crypto-ransomware in IoT networks based on energy consumption footprint / A.Azmoodeh, A. Dehghantanha, M. Conti, K. Choo. // J. Ambient Intell. Hum. Comput.. – 2018. – C. 1141–1152.
8. Eilam E. Reversing: secrets of reverse engineering / E. Eilam, E. Chikofsky. – Indianapolis: Wiley, 2005. – 624 P.
9. Elisan C. Malware, Rootkits & Botnets A Beginner's Guide / Christopher Elisan. – New York: McGraw-Hill, 2013. – 432 P.

10. Hosseini R. A state-of-the-art survey of malware detection approaches using data mining techniques / R. Hosseini, A. Souri. // Human-centric Computing and Information Sciences. – 2018. – P. 3.
11. Internet Security Report - Q1 2022 2022 [Electronic resource]. – 2022. – Resource access mode: https://www.watchguard.com/wgrd-resource-center/security-report-q1-2022
12. IoT Attacks Escalating with a 217.5% Increase in Volume [Electronic resource]. – 2022. – Resource access mode: https://www.bleepingcomputer.com/news/security/iot-attacks-escalating-with-a-2175-percent-increase-in-volume/
13. IT threat evolution in Q2 2022. Non-mobile statistics [Electronic resource]. – 2022. – Resource access mode: https://securelist.com/it-threat-evolution-in-q2-2022-non-mobile-statistics/107133/
14. Lightweight classification of IoT malware based on image recognition / [J. Su, D. Vasconcellos, S. Prasad та ін.]. // Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC). – 2018.
15. Malcaps: A Capsule Network Based Model For The Malware Classification / Zhang, Xiaoliang, Wu et al.]. // Processes. – 2021. – P. 929–947.
16. Malin C. Malware forensics: investigating and analyzing malicious code / C. Malin, E. Casey, J. Aquilina., 2008. – 132 P.
17. Margaret Rouse Malware (malicious software) [Electronic resource]. – 2022. – Resource access mode: https://searchsecurity.techtarget.com/definition/malware
18. Moser A. Limits of Static Analysis for Malware Detection / A. Moser, C. Kruegel, E. Kirda. // Proceedings of the IEEE 23rd Annual Computer Security Applications Conference. – 2007. – P. 421–430.
19. Operating System Market Share Worldwide - September 2022 [Electronic resource]. – 2022. – Resource access mode: https://gs.statcounter.com/os-market-share
20. Sikorski M. Practical malware analysis. The Hands-On Guide to Dissecting Malicious Software / M. Sikorski, A. Honig. – San Francisco: No Starch Press, 2012. – 8002 P.
21. The Antivirus Hacker's Handbook / J. Koret, E. Bachaalany – Indianapolis: John Wiley & Sons, Inc, 2015. – 384 P.
22. The Five Most Popular Operating Systems for the Internet of Things [Electronic resource]. – 2022. – Resource access mode: https://opensourceforu.com/2019/10/the-five-most-popular-operating-systems-for-the-internet-of-things/
23. The Ultimate List Of Cyber Security Statistics For 2022 [Electronic resource]. – 2022. – Resource access mode: https://purplesec.us/resources/cyber-security-statistics/
24. Thomas F. Adware: The Only Book You'll Ever Need / Thomas., 2015. – 69 P.
25. Total amount of malware and PUA [Electronic resource]. – 2022. – Resource access mode: https://portal.av-atlas.org/malware
26. What is a Backdoor Virus? - Definition, Removal & Example [Electronic resource]. – 2022. – Resource access mode: https://study.com/academy/lesson/what-is-a-backdoor-virus-definition-removal-example.html
27. Zuo Z. On the time complexity of computer viruses / Z. Zuo, Q. Zhu, M. Zhu. // IEEE Transactions on Information Theory. – 2005. – №51. – P. 2962–2966.