International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
***Article Received:** 10 July 2023 **Revised:** 31 August 2023 **Accepted:** 18 September 2023*

Malware Detection and Analysis Tools

Rajesh Yadav[1], Digvijay Singh[2]
[1]University of Southampton Malaysia, Malaysia
[2]BML Munjal University, India

ABSTRACT
The huge amounts of data and information that need to be analyzed for possible malicious intent are one ofthe big and significant challenges that the Web faces today. Malicious software, also referred to as malware developed by attackers, ispolymorphic and metamorphic in nature which can modify the code as it spreads.In addition, the diversity and volume of their variants severely undermine the effectiveness of traditional defenses that typically use signature-based techniques andare unable to detect malicious executables previously unknown. Malware family variants share typical patterns of behaviorthat indicate their origin and purpose. The behavioral trends observed either statically or dynamically can be manipulatedby usingmachine learning techniques to identify and classify unknown malware into their established families. Thissurvey paper gives an overview of the malware detection and analysis techniques and tools.

Keywords-Malware, Detection, Analysis, Tools, Machine Learning.

1. INTRODUCTION

Malware is any malicious software used to interrupt machine activity, capture sensitive information, or obtain access to private computer systems. Malware is characterized by its malicious in- tent, which works against the computer user's specifications, and does not include software whichcauses unintended harm due to a deficiency. The term badware is sometimes used and appliedto both true (malicious) malware and unintentionally harmful software. Spreadingof malware has affected everyday life, from e-governance [1] to social networks [2], from digitalautomation [3] spreading up to mobile networks [4]. Malware comes in a wide range of variationslike Virus, Worm, Trojan-horse, Rootkit, Backdoor, Botnet, Spyware, Adware etc. These classesof malware are not mutually exclusive meaning thereby that a particular malware may reveal thecharacteristics of multiple classes at the same time. In order to evade detection, malware authors introduce polymorphism to the malicious components. This means that malicious files belongingto the same malware "family", with the same forms of malicious behavior, are constantly modified and/or obfuscated using various tactics, such that they look like many different files.

Malwareis one of the most terrible and major security threats facing the Internet today. According to a survey [5], conducted by Symantec in February 2019, 47% of the organizations experienced mal- ware security incidents/network breaches in the past one year, as depicted in figures 1 and 2. Themalware is continuously growing in volume (growing threat landscape), variety (innovative mali-cious methods) and velocity (fluidity of threats). These are evolving, becoming more sophisticated and using new ways to target computers and mobile devices. McAfee [6] catalogs over 100,000 new malware samples every day means about 69 new threats every minute or about one threat persecond. With the increase in readily available and sophisticated tools, the new generation cyber threats/attacks are becoming more targeted, persistent and unknown. The advanced malware is tar- geted, unknown, stealthy, personalized and zero-day as compared to the traditional malware which was broad, known,open and one time. Once inside, they hide, replicate and disable host protec- tions. After getting installed, they call their command and control servers for further instructions, which could be to steal data, infect other machines, and allow reconnaissance. Attackers exploit vulnerabilities in web services, browsers, and operating systems, or use social engineering tech- niques to make users run the malicious code in order to spread malware. Malware authors use obfuscation techniques [7] like dead code insertion, register reassignment, subroutine reordering,instruction substitution, code transposition, and code integration to evade detection by traditional defenses like firewalls, antivirus and gateways which typically use signature based techniques andare unable to detect the previously unseen malicious executables. Commercial antivirus vendors are not able to offer immediate protection for zero-day malware as they need to analyze these to create their signatures. To overcome the limitation of signature-based methods, malware analysistechniques are being followed, which can

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
***Article Received:** 10 July 2023 **Revised:** 31 August 2023 **Accepted:** 18 September 2023*

be either static or dynamic. The malware analysis techniques help the analysts to understand the risks and intentions associated with a malicious code sample. The insight so obtained can be used to react to new trends in malware development or take preventive measures to cope with the threats coming in the future. Features derived from analysisof malware can be used to group unknown malware and classify them into their existing families.

This paper presents a review of techniques/approaches and tools for detecting and analyzingthe malware executables. There has been some study performed on comparison of static, dy- namic, and hybrid analysis for malware detection [8], whereas some researchers tried to bridge thestatic/dynamic gap [9]. Mobile technology in healthcare has also been a target of malware [10]. Few recent studies have been done on static and dynamic analysis of Android malware [11], de- tection using permission [12–14], based on system call sequences and LSTM [15].Many studies use static analysis for malware detection using exact decompilation [16], similarity testing framework [17], based on register contents [18], using two-dimensional binary program features [19], subroutine based detection [20], statistics of assembly instructions [21], file rela- tion graphs [22], de- anonymizing programmers via code stylometry [23], based upon a wavelet package technique [24], analysis and comparison of disassemblers for opcode [25].The studies that use dynamic analysis perform synthesis the semantics of obfuscated code [7], multi-hypothesis testing [26], analyzing quantitative data flow graph metrics[27], using simplified data-dependent api call graph [28], downloader graph analytics [29], access behavior [30, 31], APIs in initial behavior [32], log-based crowdsourcing analysis [33].

There have been many studies on the detection and analysis of malware using machine learning that study fine-grained features [34], deep learning [35–37], dynamic features [38], static fea- tures [36, 39], concept drift [40], predicting signatures [41], hybrid framework [42], malware metadata [43], reverse engineering of large datasets of binaries [44].

- **Techniques and tools for detecting and analyzing malware**. This paper provides the firstcomprehensive survey on techniques and tools for detecting and analyzing malware. Therehave been numerous surveys in the area of malware detection specific to machine learning,android and a few surveys on static and dynamic analysis. However, none of the work addresses the techniques and available tools.

- **State-of-the-art Survey**. This paper reveals that the most existing surveys in this area are either outdated [45] or fail to provide a holistic view of the problem, since they usually focus on a specific subset of the standard [46].

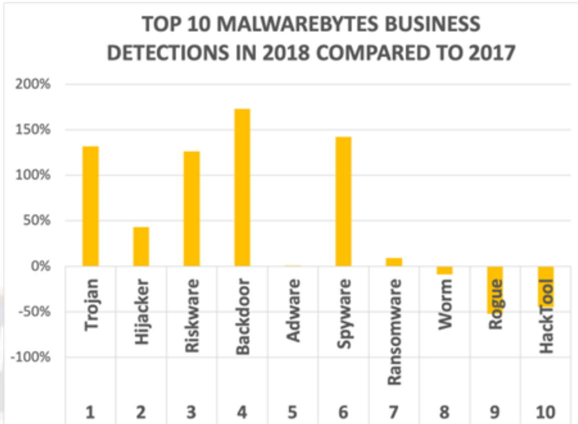

Figure 1. Top 10 malwarebytes business detections in 2018 compared to 2017 [47]

- **List of comprehensive tools**. This paper presents a novel overview of the list of com- prehensive tools available for malware detection, memory forensics, packet analysis, scan-ners/sandboxes, reverse engineering, debugging, and website analysis. Further, it differen- tiates the malware analysis tools based on specific domain and approach.

- **Guide for malware analysts**. Finally, it is realized that the contribution claimed in this paper will help, guide and assist researchers and malware analysts on getting appropriate tools for their domain-specific analysis.

The rest of the paper is organized as follows. Section II describes the different types of malware.Section III describes the ways of malware analysis. Section IV discusses malware analysis tools. Finally, Section V concludes the paper with a highlight on the scope of future work.

2. DIFFERENT TYPES OF MALWARE

With so many different types of malware – and the vast range of malicious software programs within each type

– it's important that every malware item can be unambiguously classified and easily distinguished from other malicious programs. The term malware includes viruses, worms,Trojan Horses, rootkits, spyware, keyloggers and more. To get an overview of the difference between all these types of threats and the way they work, it makes sense to divide them into groups:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
Article Received: *10 July 2023* ***Revised:*** *31 August 2023* ***Accepted:*** *18 September 2023*

2.1. Viruses and worms: the contagious threat

Viruses and worms both are designed to spread without the user's knowledge. A computer virus isa small program written to alter the way a computer operates, without the permission or knowledgeof the user. A virus must meet two criteria [8]:

1. It must execute itself. It will often place its own code in the path of execution of anotherprogram.

2. It must replicate itself. For example, it may replace other executable files with a copy of thevirus-infected file.

Viruses can infect desktop computers and network servers alike. Some viruses are programmed todamage the computer by damaging programs, deleting files, or reformatting the hard disk. Othersare not designed to do any damage, but simply to replicate themselves and make their presence known by presenting text, video, and audio messages. Even these benign viruses can create prob-lems for the computer user. They typically take up computer memory used by legitimate programs. As a result, they often cause erratic behavior and can result in system crashes. Also, many virusesare bug-ridden, and these bugs may lead to system crashes and data loss.Computer worms, on the other hand, spread across the internet by replicating itself on comput-ers via their network. Both viruses and worms can carry a so-called "payload", malicious code designed to do damage. Worms are programs that replicate themselves from system to system without the use of a host file. This is in contrast to viruses, which requires the spreading of an infected host file. Although worms generally exist inside of other files, often Word or Excel doc-uments, there is a difference between how worms and viruses use the host file. Usually the worm will release a document that already has the "worm" macro inside the document. The entire doc-ument will travel from computer to computer, so the entire document should be considered the worm. PrettyPark is a particularly prevalent example.

2.2. Trojans and Rootkits: the masked threat

Trojans and rootkits are grouped as they both seek to conceal attacks on computers. Trojan Horsesare malignant pieces of software pretending to be benign applications. Users therefore downloadthem thinking they will get a useful piece of software and instead end up with a malware infectedcomputer. A Trojan horse, or Trojan, in computing is a generally non-self-replicating type of malware program containing malicious code that, when executed, carries out actions determinedby the nature of the Trojan, typically causing loss or theft of data, and possible system harm. Theterm is derived from the story of the wooden horse used to trick defenders of Troy into taking concealed warriors into their city in ancient Anatolia, because computer Trojans often employ a form of social engineering.

Rootkits are different. They are a masking technique for malware, but do not contain damaging software. Rootkit techniques were invented by virus writers to conceal malware, so it could go unnoticed by antivirus detection and removal programs. A rootkit is a stealthy type of software, typically malicious, designed to hide the existence of certain processes or programs from normalmethods of detection and enable continued privileged access to a computer. The term rootkit is aconcatenation of "root" (the traditional name of the privileged account on Unix operating systems)and the word "kit" (which refers to the software components that implement the tool).

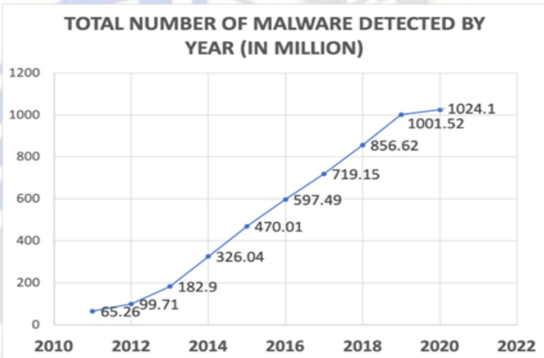

Figure 2. Total number of malware detected by year (in million) [48]

2.3. Spyware and keyloggers: the financial threat

Spyware and keyloggers are malware used in malicious attacks like identity theft, phishing and social engineering, threats designed to steal money from unknowing computer users, businesses and banks. Spyware is a type of malicious software (also called "malware") that scammers try toinstall on your computer. As the name suggests, spyware programs allow people to spy on what you are doing on your computer: the websites you visit, the files you use and the details you storeon your PC

Key-loggers are a particular type of spyware. Key-loggers secretly record what keys you press on your keyboard and sends this data back to the scammer over the internet. Scammers use theseprograms to steal passwords such as online banking passwords. They may also use spyware to steal other personal information from you such as documents that you have stored on your computer. Scammers use a wide range of tricks to get their spyware and key-loggers loaded on to your computer. This usually involves tricking you into clicking on a link in a spam email they have sent, or visiting a website that they have set up solely to infect people's computers. Other sources of spyware and key-loggers are free games or music that you can download from the internet. When they are delivered in this way, they are sometimes called "Trojans"–a file that claims to befor some harmless purpose so it can get under your guard, but contains a nasty surprise.

3. MALWARE ANALYSIS

Before creating the signatures for newly arrived malware, these are required to be analyzed to understand the associated risks and intentions. The malicious program and its capabilities can beobserved either by examiningits code or by executing it in a safe environment.

3.1. Static analysis

Analyzing malicious software without executing it is called static analysis. The detection patternsused in staticanalysis include string signature, byte-sequence n-grams, syntactic library call, con-trol flow graph and opcode (operational code) frequency distribution etc. The executable has to be unpacked and decrypted before doing static analysis. The disassembler/debugger and memorydumper tools can be used to reverse com pile windows executables. Disassemble/Debugger toolslike IDA Pro and OllyDbg displays the malware's code as Intel X86 assembly instructions, whichprovide a lot of insight into what the malware is doing and provide patterns to identify the attack-ers. Memory dumper tools like LordPE [9] and OllyDump [10] are used to obtain protected codelocated in the system's memory and dump it to a file. This is a useful technique to analyze packedexecutables which are difficult to disassemble. Binary obfuscation techniques, which transform the malware binaries into self-compressed and uniquely structured binary files, are designed to resist reverse engineering and thus make the static analysis very expensive and unreliable. More-over, when utilizing binary executables (obtained by compiling source code) for static analysis, the information like size of data structures or variables gets lost thereby complicating the malwarecode analysis [11]. The evolving evasion techniques being used by malware writers to thwart static analysis led to the development of dynamic analysis.

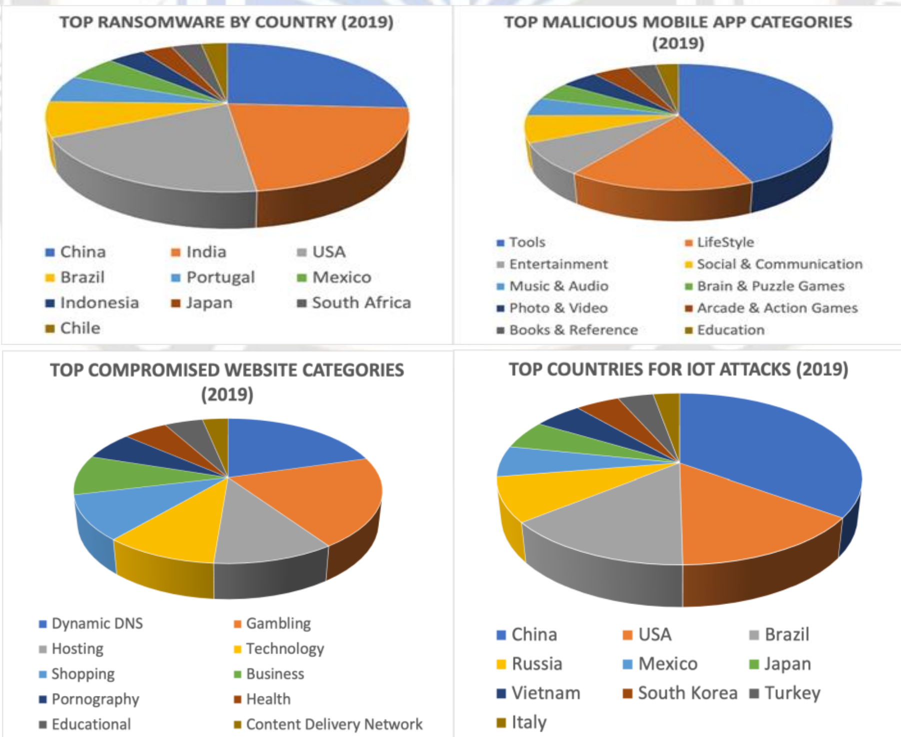

Figure 3. (a) Top ransomware by country (2019) (b) Top malicious mobile app categories (2019) (c) Topcompromised website categories (2019) (d) Top countries for IoT attacks (2019) [5]

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
***Article Received:** 10 July 2023 **Revised:** 31 August 2023 **Accepted:** 18 September 2023*

Moser et al. [12], explored the drawbacks ofstatic analysis methodology. In their work, they introduced a scheme based on code obfuscation revealing the fact that the static analysis alone is not enough to detect or classify malware. Fur- ther, they proposed that dynamic analysis is a necessary complement to static analysis as it is less vulnerable to code obfuscation conversion.

3.2. Dynamic analysis

Analyzing the behavior of a malicious code (interaction with the system) while it is being exe- cuted in a controlled environment (virtual machine, simulator, emulator, sandbox etc.) is called dynamic analysis. Before executing the malware sample, the appropriate monitoring tools like Process Monitor [13] and Capture BAT [14](for file system and registry monitoring), Process Ex- plorer [15] and Process Hackerreplace [16] (for process monitoring), Wireshark [17] (for network monitoring) and Regshot [18] (for system change detection) are installed and activated. Vari- ous techniques that can be applied to perform dynamic analysis include function call monitoring,function parameter analysis, information flow tracking, instruction traces and autostart extensi-bility points etc. [11]. Dynamic analysis is more effective as compared to static analysis and does not require the executable to be disassembled. It discloses the malware's natural behavior which is more resilient to static analysis. However, it is time intensive and resource consuming, thus elevating the scalability issues. The virtual environment in which malware are executed is different from the real one and the malware may perform in different ways resulting in artificial behavior rather than the exact one. In addition to this, sometimes the malware behavior is trig- gered only under certain conditions (on specific system date or via a specific command) and can'tbe detected in virtual environment. Several online automated tools exist for dynamic analysis of malware,

e.g. Norman Sandbox [19], CWSandbox [20], Anubis [21] and TTAnalyzer [22], Ether[23] and ThreatExpert [24]. The analysis reports generated by these tools give in-depth under- standing of the malware behavior and valuable insight into the actions performed by them. The analysis system is required to have an appropriate representation for malware, which are then usedfor classification either based on similarity measure or feature vectors. However a large number of new malware samples arriving at anti-virus vendors every day requires an automated approach to limit the number of samples that require close human analysis. Several Artificial Intelligence techniques, particularly machine-learning based techniques have been used in the literature for automated malware analysis and classification.

4. MALWARE ANALYSIS TOOLS

Analysts use tools for analyzing malware to protect and predict future attacks, and share knowledge among them. Open source tools are often the first choice to carry out such actions. It's no secret that distributing malware is a big business and the fast-growing malware epidemic will onlygrow in ability and efficiency in the years to come. Using open source malware analysis tools, researchers will check, identify and log different variants of malicious triggers when analyzing thelife-cycle of attack. As malware trading forums are proliferating on the dark web, the crypters, botnets and zero-days needed to carry out powerful attacks have become easier than ever to get. With the growth of complexity of malware variants, the jobs of understanding and benchmarkingthe specific type have become harder. It's the job of security researchers and analysts to find outthe right tool to analyze each specific type of attack. We now present some open source malwareanalysis tools that can help the researchers and security engineers.

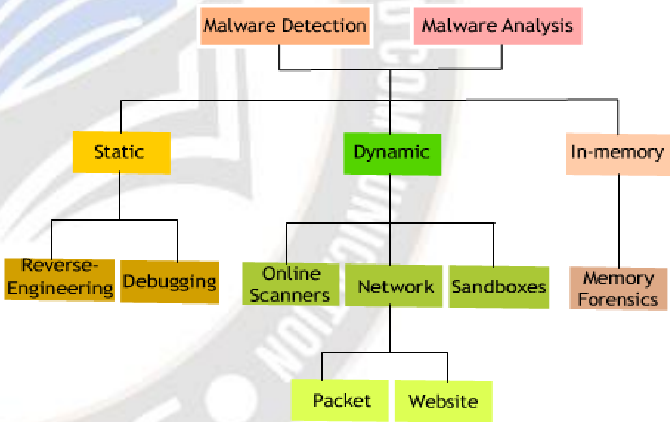

Figure 4. Malware Detection and Analysis Tools

4.1. Open Source Malware Analysis Tools

Google Rapid Response (GRR). The GRR platform is an incident response system developed by security researchers at Google, identifying common malware footprints workstations focusedon remote live forensics. This consists of an application that is installed on the target system to communicate with the agent and a server infrastructure. GRR is a python client (agent) that is installed on target systems, and python server infrastructure that can manage and talk to clients. Once both the server side and the agent are deployed they can become GRR clients and start receiving messages from the servers. Then the incident response staff on the host computer will perform various technical operations, such as reviewing the memory, looking for different settingsand handling software choices. GRR has been designed to run on a scale so analysts can easily capture and process data from large numbers of computers. GRR's goal is to support forensics and investigations in a simple, flexible way that allows investigators to rapidly triage incidents andconduct remote analysis.

Cuckoo Sandbox. Created by a team of volunteers during the Google Summer of Code initiativeback in 2010, it is an open source framework that automates malicious file analysis for Windows,OS X, Linux and Android and offers comprehensive and practical input on how each presented file operates in isolated environments. And since it is open source software, developers are constantlywriting plugins that provide enhanced features. Cuckoo is used by malware detection and securityfirms to help ease the strain of manually wading through troves of potentially malicious data. Themodular design allows the recording and analysis phases simple to configure. In recent years, it has, understandably, become one of the most commonly used open source tools. In 2012, Cuckoopublished Malwr, a sandbox-as - a-service that allows users to use their collected data through aneasy-to-use GUI. The goal was to act as an option for users who can't handle Cuckoo properly butstill want to exploit their intellect.

Yara Rules. Another open source malware identification tool that can identify samples of malware based on textual or binary trends once they are tested in Cuckoo. Investigators use Yara to com- pose pattern-based definitions of the malware families. YARA stands for "Yet Another RecursiveAcronym" as the descriptions are called rules. This helps researchers to identify and categorize apparently similar malware types and can be adapted for use inside Cuckoo. IBM calls Yara the "pattern matching Swiss army knife" of the malware researcher and can be used on both Windowsand Linux computers.

Table 1 list some tools available for malware detection, memory forensics, packet analysis, scanners/sandboxes, reverse engineering, debugging, and website analysis.

4.2. Mobile Malware Analysis Tools

APKTool. A tool for reverse engineering 3rd party, closed, binary Android apps. By making several changes, it can decode resources to almost original form and restore them. It also makes it easier to deal with a device owing to the project such as file creation and completion of some repet-itive tasks such as creating the apk etc.We can decode APK resources to almost original form with the help of Apktool; we can modify the source code on the fly and rebuild the decoded resources back into APK. Its project-like structure makes working with them easy. Apktool can decode APK resources (resources.arsc, classes.dex and XMLs), rebuild decoded resources back to binary APK, organize and handle APKs that depend on framework resources along with automating the repetitive tasks.

Dex2Jar. Dex2Jar is a free tool for dealing with the files Android ".dex" and Java ".class." An- droid programs are assembled into".dex" (Dalvik Executable) scripts, which in effect are zippedonto the computer into a single.apk file. Android will automatically create the ".dex" folders, byconverting the compiled applications written in the Java. Dex2Jar reads the dex instruction to dex-ir format and can convert to ASM format. It can alsobe used to perform some basic deobfuscation. Dex2Jar's core feature is converting an APK classes.dex file to classes.jar, or vice versa. So, using any Java decompiler, it is possible to view the source code of an Android application, and it is fully legible.

Table 1. Tools available for malware detection, memory forensics, packet analysis, scanners and sandboxes.

Detection Tools		**scanners and sandboxes**	
AnalyzPE	Wrapper for a variety of tools for reporting onPE files.	W AndroTotal	Online analysis of APKs against multiple mobileantivirus apps
chkrootkit	Linux rootkit detector.	APK Analyzer	Dynamic analysis of APKs
Detect-It-Easy	A program for determining types of files.	AVCaesar	Online scanner and malware repository
hashdeep	Compute digest hashes with a variety of algo-rithm	Cryptam	Analyze suspicious office documents
Loki	Host based scanner for IOCs.	Cuckoo Sandbox	Open source sandbox and automated analysis sys-tem

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
***Article Received:** 10 July 2023 **Revised:** 31 August 2023 **Accepted:** 18 September 2023*

MASTIFF	Static analysis framework.	Comodo Valkyrie	File verdict system that conducts several analysis using run-time and hundreds of features from a file
TrID	File identifier.	firmware.re	Unpacks, scans and analyzes firmware packages
YARA	Pattern matching tool for analysts.	Hybrid Analysis	Online malware analysis tool
Memory Forensics Tools		IRMA	An asynchronous and customizable analysis plat-form for suspicious files
DAMM	Differential Analysis of Malware in Memory, builto Volatility	Joe Sandbox	Deep malware analysis.
evolve	Web interface for the Volatility Memory Forensics Framework	Jotti	Online AV scanner
FindAES	Find AES encryption keys in memory	Limon	Sandbox for Analyzing Linux Malware
Packet Analysis Tools		NVISO ApkScan	Dynamic analysis of APKs
Network Miner	A Network Forensic Analysis Tool (NFAT) for Windows	PDF Examiner	Analyse suspicious PDF files
NetworkTotal	Online analysis of .pcap files to detec viruses,worm trojans and malware.	SEE	"Sandboxed Execution Environment", a frame- work for building test automation in secured environments
PacketTotal	Online engine for analyzing .pcap files and visu- al network traffic within, useful for mal- ware anal incident response. My review	i URL Analyzer	Dynamic analysis of URL files
Wireshark	Widely-used network protocol analyzer.	VirusTotal	Online analysis of malware samples and URLs

Mobile-Sandbox. Mobile-Sandbox provides static and dynamic malware analysis for Android OS smartphones. The system is designed to automatically evaluate Android software in two novelways: (1) by integrating static and dynamic analysis, i.e., static analysis findings are used to di- rect dynamic analysis and expand coverage of executed code, and (2) by using different logging methods for native API calls. It can evaluate the application with different modules within the static analysis component to get a summary of the program. To achieve this, it uses the VirusTotal service to perform several anti-virus scans, parse the manifest file and finally decompile the ap- plication to better identify suspect code. Within the dynamic analysis, it can run the application in an emulator and log every application operation, i.e. it logs both actions performed in the JavaVirtual Machine Dalvik and actions performed in native libraries that may be bundled with the application.

4.3. Other Analysis Tools

Malzilla. Malzilla is a useful malware hunting tool for analyzing websites containing malicious code. Web pages that contain exploits often use a sequence of redirects and obfuscated codeto make it difficult for someone to track them. This allows users to access websites and obtain all of their source code, such as wget, without visiting the site and potentially damaging their device. This program has the option of switching the user agents and picking the user's referrer. This shows the full list of webpages for browsers and all the headers for HTTP. It also has proxyfeatures, complex decoders and, most notably, JavaScript code deobfuscation, all in one program.

Wireshark. Wireshark, a network monitoring application once known as Ethereal, records pack-ets and shows them in the human-readable format in real-time. It intercepts traffic and transformsthe binary data into a readable format for users. Wireshark includes filters, color coding, and otherfeatures that allow individuals to dig deep into network traffic and inspect individual packets. It isthe leading network traffic analyzer in the world, and an essential tool for any skilled security ordevice administrator. This free software allows people to track networktraffic in real-time, and isoften the best tool on any network for troubleshooting problems. Common issues that Wiresharkcan deal with troubleshooting include lost messages, latency issues and malicious network opera-tion. It enables network data to be held under a microscope and offers resources for filtering anddigging into that information, zooming into the root cause of the issue. It is used by management to detect defective network equipment that loses packets, latency problems caused by machines transmitting traffic around the world and data exfiltration or even intrusion attempts against any entity.

VirusTotal. Virustotal is a service that analyzes suspicious files and URLs and helps to detect viruses, worms, Trojans and all kinds of malware detected by antivirus engines quickly. In addi-tion to a variety of methods for removing signals from the studied material, VirusTotal inspects products with over 70 antivirus scanners and URL / domain blacklisting services. Every person can use their browser to pick a file from their device, and submit it to VirusTotal. VirusTotal offersvarious methods for uploading data, including the default public web portal,

desktop uploaders, browser extension, and a programmatic API. The web interface has the greatest scanning priorityamong the forms of application which are available to the public. The specifications can be madeusing the HTTP-based public API in any programming language. It also offers a variety of otherfunctions, including the VirusTotal Community: a network that allows users to report on files andURLs and exchange comments with each other. This can be helpful in detecting malicious contentand also in finding false positives – regular and harmless objects identified as dangerous by one ormore scanners.

5. CONCLUSION

This survey paper presents a summary of malware detection and analysis techniques and tools. Inparticular, the different tools available for malware detection, memory forensics, packet inspec- tion, scanners/sandboxes, reverse engineering, hacking, and website analysis have been thrown light. Since most of the current surveys typically concentrate on a specific subset of the model, this paper offers an in-depth study of methods to identify and evaluate malware with a clear understanding of domain-specific analytics.

REFERENCES

[1] S. K. Talukder, M. I. I. Sakib, and M. M. Rahman, “Model for e-government in bangladesh:A unique id based approach,” in 2014 International Conference on Informatics, ElectronicsVision (ICIEV), pp. 1–6, May 2014.

[2] S. Talukder and B. Carbunar, “When friend becomes abuser: Evidence of friend abuse in facebook,” in Proceedings of the 9th ACM Conference on Web Science, WebSci ’17, (NewYork, NY, USA), ACM, June 2017.

[3] S. K. Talukder, M. I. I. Sakib, and M. M. Rahman, “Digital land management system: A newinitiative for bangladesh,” in 2014 International Conference on Electrical Engineering andInformation Communication Technology, pp. 1–6, April 2014.

[4] S. Talukder, I. I. Sakib, F. Hossen, Z. R. Talukder, and S. Hossain, “Attacks and defenses inmobile ip: Modeling with stochastic game petri net,” in 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), pp. 18– 23, IEEE, 2017.

[5] S. Corporation, “Internet security threat report.” Symantec, shorturl.at/aqKO7, 2019.

[6] M. Labs, “Mcafee labs threats reports.” McAfee, shorturl.at/izEJU, 2019.

[7] T. Blazytko, M. Contag, C. Aschermann, and T. Holz, “Syntia: Synthesizing the semantics of obfuscated code,” in 26th {USENIX} Security Symposium ({USENIX} Security 17), pp. 643– 659, 2017.

[8] A. Damodaran, F. Di Troia, C. A. Visaggio, T. H. Austin, and M. Stamp, “A comparison ofstatic, dynamic, and hybrid analysis for malware detection,” Journal of Computer Virologyand Hacking Techniques, vol. 13, no.1, pp. 1–12, 2017.

[9] B. Anderson, C. Storlie, and T. Lane, “Improving malware classification: bridging the static/dynamic gap,” in Proceedings of the 5th ACM workshop on Security and artificial intelligence, pp. 3–14, 2012.

[10] S. Talukder, S. Witherspoon, K. Srivastava, and R. Thompson, “Mobile technology in healthcare environment: Security vulnerabilities and countermeasures,” arXiv preprint arXiv:1807.11086, 2018.

[11] C. Raghuraman, S. Suresh, S. Shivshankar, and R. Chapaneri, “Static and dynamic malwareanalysis using machine learning,” in First International Conference on Sustainable Tech- nologies for Computational

Intelligence, pp. 793–806, Springer, 2020.

[12] A. Arora, S. K. Peddoju, and M. Conti, “Permpair: Android malware detection using per- mission pairs,” IEEE Transactions on Information Forensics and Security, 2019.

[13] M. Alazab, M. Alazab, A. Shalaginov, A. Mesleh, and A. Awajan, “Intelligent mobile mal-ware detection using permission requests and api calls,” Future Generation Computer Sys-tems, 2020.

[14] S. Talukder and B. Carbunar, “Abusniff: Automatic detection and defenses against abu- sive facebook friends,” in Twelfth International AAAI Conference on Web and Social Media,2018.

[15] X. Xiao, S. Zhang, F. Mercaldo, G. Hu, and A. K. Sangaiah, “Android malware detection based on systemcall sequences and lstm,” Multimedia Tools and Applications, vol. 78, no. 4, pp. 3979–3999, 2019.

[16] E. Schulte, J. Ruchti, M. Noonan, D. Ciarletta, and A. Loginov, “Evolving exact decompila- tion,” in Workshop on Binary Analysis Research (BAR), 2018.

[17] J. Upchurch and X. Zhou, “Variant: a malware similarity testing framework,” in 2015 10thInternational Conference on Malicious and Unwanted Software (MALWARE), pp. 31–39, IEEE, 2015.

[18] M. Ghiasi, A. Sami, and Z. Salehi, “Dynamic vsa: a framework for malware detection basedon register contents,” Engineering Applications of Artificial Intelligence, vol. 44, pp. 111– 122, 2015.

[19] J. Saxe and K. Berlin, “Deep neural network based malware detection using two dimen- sional binary program features,” in 2015 10th International Conference on Malicious and Unwanted Software (MALWARE),pp. 11–20, IEEE, 2015.

[20] J. Sexton, C. Storlie, and B. Anderson, “Subroutine based detection of apt malware,” Journalof Computer Virology and Hacking Techniques, vol. 12, no. 4, pp. 225–233, 2016.

[21] P. Khodamoradi, M. Fazlali, F. Mardukhi, and M. Nosrati, “Heuristic metamorphic malware detection based on statistics of assembly instructions using classification algorithms,” in 2015 18th CSI International Symposium on Computer Architecture and Digital Systems (CADS),pp. 1–6, IEEE, 2015.

[22] L. Chen, T. Li, M. Abdulhayoglu, and Y. Ye, “Intelligent malware detection based on file relation graphs,” in Proceedings of the 2015 IEEE 9th International Conference on SemanticComputing (IEEE ICSC 2015), pp. 85–92, IEEE, 2015.

[23] A. Caliskan-Islam, R. Harang, A. Liu, A. Narayanan, C. Voss, F. Yamaguchi, and R. Green-stadt, “De-anonymizing programmers via code stylometry,” in 24th {USENIX} Security Sym- posium ({USENIX} Security 15), pp. 255–270, 2015.

[24] B. Gu, Y. Fang, P. Jia, L. Liu, L. Zhang, and M. Wang, “A new static detection method of malicious document based on wavelet package analysis,” in 2015 International Conferenceon Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), pp. 333– 336, IEEE, 2015.

[25] M. Nar, A. G. Kakisim, M. N. Yavuz, and İ. Soğ ukpinar, “Analysis and comparison of dis- assemblers for opcode based malware analysis,” in 2019 4th International Conference on Computer Science and Engineering (UBMK), pp. 17–22, IEEE, 2019.

[26] P. Vadrevu and R. Perdisci, “Maxs: Scaling malware execution with sequential multi- hypothesis testing,” in Proceedings of the 11th ACM on Asia Conference on Computer andCommunications Security, pp. 771–782,2016.

[27] T. Wüchner, M. Ochoa, and A. Pretschner, “Robust and effective malware detection through quantitative data flow graph metrics,” in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 98–118, Springer, 2015.

[28] A. A. E. Elhadi, M. A. Maarof, and B. Barry, “Improving the detection of malware behaviourusing simplified data dependent api call graph,” International Journal of Security and Its Applications, vol. 7, no. 5, pp. 29–42, 2013.

[29] B. J. Kwon, J. Mondal, J. Jang, L. Bilge, and T. Dumitras,, “The dropper effect: Insights into malware distribution with downloader graph analytics,” in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1118–1129, 2015.

[30] A. Mohaisen, O. Alrawi, and M. Mohaisen, “Amal: High-fidelity, behavior-based automated malware analysis and classification,” computers & security, vol. 52, pp. 251–266, 2015.

[31] W. Mao, Z. Cai, D. Towsley, and X. Guan, “Probabilistic inference on integrity for access behavior based malware detection,” in International Symposium on Recent Advances in In-trusion Detection, pp. 155–176, Springer, 2015.

[32] N. Kawaguchi and K. Omote, “Malware function classification using apis in initial behavior,”in 2015 10th Asia Joint Conference on Information Security, pp. 138–144, IEEE, 2015.

[33] A. Raff, D. Peri, and A. Lotem, "System and methods for malware detection using log basedcrowdsourcing analysis," Aug. 27 2019. US Patent 10,397,246.

[34] X. Jiang, B. Mao, J. Guan, and X. Huang, "Android malware detection using fine-grained features," Scientific Programming, vol. 2020, 2020.

[35] M. K. Alzaylaee, S. Y. Yerima, and S. Sezer, "Dl-droid: Deep learning based android mal-ware detection using real devices," Computers & Security, vol. 89, p. 101663, 2020.

[36] S. Talukder, "Tools and techniques for malware detection and analysis," arXiv preprint arXiv:2002.06819, 2020.

[37] Z. Yuan, Y. Lu, and Y. Xue, "Droiddetector: android malware characterization and detection using deep learning," Tsinghua Science and Technology, vol. 21, no. 1, pp. 114–123, 2016.

[38] I. Sogukpinar, "Analysis and evaluation of dynamic feature-based malware detection meth- ods," in Innovative Security Solutions for Information Technology and Communications: 11th International Conference, SecITC 2018, Bucharest, Romania, November 8–9, 2018, Revised Selected Papers, vol. 11359, p. 247, Springer,2019.

[39] G. Laurenza, L. Aniello, R. Lazzeretti, and R. Baldoni, "Malware triage based on static features and public apt reports," in International Conference on Cyber Security Cryptography and Machine Learning, pp. 288–305,Springer, 2017.

[40] R. Jordaney, K. Sharad, S. K. Dash, Z. Wang, D. Papini, I. Nouretdinov, and L. Cavallaro, "Transcend: Detecting concept drift in malware classification models," in 26th {USENIX}Security Symposium ({USENIX}Security 17), pp. 625–642, 2017.

[41] M. Howard, A. Pfeffer, M. Dalai, and M. Reposa, "Predicting signatures of future mal- ware variants," in 2017 12th International Conference on Malicious and Unwanted Software(MALWARE), pp. 126–132, IEEE, 2017.

[42] Z. Feng, S. Xiong, D. Cao, X. Deng, X. Wang, Y. Yang, X. Zhou, Y. Huang, and G. Wu, "Hrs: A hybrid framework for malware detection," in Proceedings of the 2015 ACM InternationalWorkshop on International Workshop on Security and Privacy Analytics, pp. 19–26, 2015.

[43] M. Asquith, "Extremely scalable storage and clustering of malware metadata," Journal of Computer Virology and Hacking Techniques, vol. 12, no. 2, pp. 49–58, 2016.

[44] M. Polino, A. Scorti, F. Maggi, and S. Zanero, "Jackdaw: Towards automatic reverse engi-neering of largedatasets of binaries," in International Conference on Detection of Intrusionsand Malware, and Vulnerability Assessment, pp. 121–143, Springer, 2015.

[45] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware- analysis techniques and tools," ACM Comput. Surv., vol. 44, Mar. 2008.

[46] D. Uppal, V. Mehra, and V. Verma, "Basic survey on malware analysis, tools and techniques," International Journal on Computational Sciences & Applications (IJCSA), vol. 4, no. 1,p. 103, 2014.

[47] M. Labs, "2019 state of malware." Malwarebytes, shorturl.at/bjtP8, 2019.

[48] I. Lapowsky, "Malware last 10 years." AV-TEST, shorturl.at/yzN01, 2020.