Detecting Remote Access Trojan (RAT) Attacks based on Different LAN Analysis Methods

Salar Jamal Rashid
Electronic Technologies Department, Mosul Technical Institute, Northern Technical University, Mosul, Iraq
salar.jamal@ntu.edu.iq (corresponding author)

Shatha A. Baker
Electronic Technologies Department, Mosul Technical Institute, Northern Technical University, Mosul, Iraq
shathaab@ntu.edu.iq

Omar I. Alsaif
Electronic Technologies Department, Mosul Technical Institute, Northern Technical University, Mosul, Iraq
omar.alsaif@ntu.edu.iq

Ali I. Ahmad
Al-Noor University, Mosul, Iraq
ali.ibrahim@alnoor.edu.iq

Received: 17 July 2024 | Revised: 1 August 2024 | Accepted: 11 August 2024

 DOI: https://doi.org/10.48084/etasr.8422

ABSTRACT

Cyberattacks aim to access confidential information or disrupt system functionality. These days, they can take the form of attacks that give the attacker complete control over the victim's computer. Remote Access Trojans (RAT) are malware designed for these purposes. RAT gives an attacker direct access to a victim's computer and allows him to interact with the victim to steal confidential information, spy on him in real time, or interact directly with him through a dialogue box. RATs are used for information theft, surveillance, and extortion of victims. This study installed multiple virtual machines as a prototype for both the attacker and the victim, interconnected on a Local Area Network (LAN). RAT installations were explored using Mega RAT version 1.5 Beta. Ultimately, various RAT attacks were executed on target machines, and a range of static and dynamic analysis tools were employed to identify RAT. The scenarios implemented on the LAN demonstrated that RATs can be built and used with ease. Furthermore, their attacks can be identified through static or dynamic analysis using various freely available tools. The findings show that the static detection approach to identify RAT malware is more user-friendly compared to dynamic methods. However, dynamic detection can be easily performed using cost-free software.

Keywords-trojan; RAT; mega RAT; RATVMWare; wireshark

I. INTRODUCTION

Online banking and shopping allow everyone to communicate and do business on the Internet. Many people use social media and store private data on their computers, laptops, and phones. Thus, information security is becoming increasingly important [1]. Spying is not new. Today, attackers use this strategy online to quickly steal user data or make money using deception. Malicious software can infect user devices through unsafe websites, custom-designed e-mails, cookies, and social engineering attacks that look like ads and may contain trojans [2-4]. There are many classes of trojans, depending on their intended use. The most common trojans are Remote Access Trojans (RATs), which allow the attacker to remotely control the target machine [5]. Users unknowingly install RATs on their computers by mistaking them for legitimate apps. Malware writers create RAT to spy, control, or damage systems, as its backdoor allows remote computer

control. To build a botnet, attackers may target competitors for sensitive data or spread RATs to vulnerable machines [1]. Due to its ability to steal confidential data and execute hacker-controlled malicious instructions, RAT is a major threat to every organization [6]. RATs present a substantial risk to network security by allowing unauthorized access and manipulation of systems. This study aims to improve network defense capabilities by exploring various LAN analysis techniques to identify RAT attacks, which are crucial risks that require effective detection methods.

In [7], the investigation and recovery from trojan attacks on the network were explored using digital forensic tools, namely FTK Imager, Wireshark, and Volatility. This study experimented with two trojan types, namely RAT and HTTP trojans, analyzing captured network packets. This study showed that Wireshark is a very useful investigation tool due to its successful detection of HTTP and RAT attacks. At the same time, both Volatility and FTK Imager can detect only a RAT attack. In [8], the effect of malware was analyzed, particularly attacks by worms and trojans. This study experimented with these attacks on PCs and detected them using the VirusTotal website, Malwarebytes, Avast Antivirus, and Wireshark. Trojans and worms can attack the PC, gaining valuable information or harming the user. In [9], various methods were presented to disinfect an infected computer and to play safe while working on the internet. This study explained how RAT can be created using Beast version 2.06 and how to bind it to another program. Some methods were also presented to protect against these types of malware. In [10], vulnerabilities were identified in Android using the Ghost framework. A vulnerability was identified in Android smartphones by exploiting the Android Debug Bridge (ADB), and the test results were analyzed to identify remote access trojan attacks. Exploitation involved connecting the testing device to ADB, exploiting it, entering ADB shell commands, and gaining remote access. The results showed that Android version 9 could be remotely accessed by entering an exploit through ADB, allowing unauthorized parties to perform activities such as opening the lock screen, accessing the system directory, and modifying the system. In [11], the track and analysis of RAT were discussed using the FTK Imager for live forensic investigations. This study aimed to improve the security of computer systems and help organizations protect their assets and data against malicious cyberattacks. The advantage of this study was the knowledge of the presence of RAT despite its removal. This method involved installing Kali Linux, conducting forensic analysis using FTK Imager, and developing and using viruses.

This study conducted a practical experiment that involved setting up a virtual environment using VMware and creating virtual machines for the attacker and the victim in a LAN. A RAT server was developed using Mega RAT version 1.5 Beta, and various RAT attacks were carried out on the victim machine. This study highlights different malware analysis methods, including static and dynamic analysis, and explores malware detection techniques, such as signature-based and behavior-based systems. Host- and network-based intrusion detection systems are distinguished, highlighting the role of each in identifying and mitigating malware threats.

II. BACKGROUND

A. Remote Access

Any technique for managing a computer from a distance is referred to as remote access. It is becoming more popular and widely used when being physically close to a system is inconvenient or difficult, or when one wants to access something on the Internet that is not available in his location. Remote access is possible to any computer connected to a LAN or the Internet. Remote access software can display the remote computer's screen on a local monitor and instantaneously send mouse and keystroke commands to the remote computer. Several Windows operating systems use graphical remote administration tools to grant access to the GUI. Terminal Services, a Windows NT feature that allows multiple concurrent interactive logon sessions, is natively supported by recent Windows operating systems. Terminal Services uses the Remote Desktop Protocol (RDP), which by default runs on TCP port 3389. Terminal services use Windows authentication to verify users when they initiate remote sessions [12].

B. Trojans

A trojan horse is a seemingly helpful program with hidden features that allow it to exploit the user privileges the program is running on, creating a security risk, and performing actions that the program's user did not expect [13]. A trojan horse can propagate by deceiving users into believing that it is a beneficial application, or it can be deliberately included by a programmer in other helpful software [14]. Trojans distinguish from viruses, as a trojan horse does not spread or reproduce by itself, needs user intervention to operate, and often includes intentionally launching the host program [15]. When running on a machine, the trojan can perform a variety of tasks, including erasing or corrupting data. Trojans are designed to inflict maximum damage on a victim or perform specific tasks [16]. Following the victim's computer infection, the trojan can provide the passwords for each email account that will be opened. Social media accounts, including Instagram, Twitter, Facebook, and all other websites, are susceptible to having their usernames and passwords altered or stolen at any time. A trojan keylogger function must be enabled to activate this feature [17]. The trojan regularly sends all keyboard inputs to the attacker using the user's email address, even when the attacker is not online. The intervals are determined by the trojan keylogger settings [18].

To allow communication, the trojan horse server will open a listening port on the compromised computer, allowing the perpetrator to establish a connection. Because of restricted privileges and to avoid conflicts with other installed programs, a non-privileged port larger than 1024 is typically used. When trojan horses first appeared, every trojan had a default connection port. By mapping the listening port to the associated malware name, one may identify an installed trojan by knowing the port number. These days, a trojan's listening port can be configured to any port an attacker requires, making it hard to identify a trojan by its port number. However, a few outdated trojans continue to operate on the Internet using their default port [19].

Trojan programmers extensively utilize TCP communication between client and server applications because it is a much simpler protocol to handle than the connectionless UDP, avoiding the trouble of having to cope with lost and rearranged UDP packets. Before activating the port, the trojans within the system are initially silent. An authentication password may be provided during communication and following the three-way handshake. This method of authentication usually uses a password hardcoded into the server, set up before the server component installation. Currently, relatively few trojans use zero-knowledge authentication methods such as challenge-response systems. Naturally, the password is only in place to prevent other users from gaining control of the victim server [15, 20].

C. RATs

RATs are programs that allow malicious attackers to take over a computer and obtain victim data by creating a backdoor in the user's system [21-23]. RATs are constantly being developed with new techniques to enable attackers to connect remotely and interact with the victim machine [24]. The primary way that a RAT infects a target is by instructing them to install a changed file [25]. This file can be distributed via a user program, such as Java Downloader, or social media sites (Facebook, Twitter, etc.). The ability of RATs to turn on microphone and camera devices at any moment is the most straightforward attack. Even when the user is not using the computer, an attacker can connect to the webcam and watch, listen, or record all of the conversations taken on in the room if the user's system is open and connected to the Internet. Furthermore, the remote access function enables the attacker to browse any website and download any file to the user's computer. When the trojan is executed, the RATs can send the attacker information about all installed applications, account passwords, application passwords, hardware, and system features. This creates a real risk and a serious threat if the user uses Internet banking or saves business data on the system [5]. A RAT consists of two files: a client file and a server file. RATs are mostly used in client-server connections. The attacker controls the client program while the server application is installed on the victim's computer. Additionally, the client could be a telnet client or a browser [15]. Activist groups or intelligence agencies also use RATs for specific purposes, such as spying and blackmail [26].

It is important to understand what artifacts may be found in an investigation. Generally, trojans/backdoors are classified into three components [14]:

- The client is used to control the backdoor from a remote location.
- The server, which is the backdoor itself, is often wrapped up in the overall trojan. It is configured with specific choices and can also contain other assistant modules, termed plugins.
- The creation toolkit configures the behavior of the backdoor before it is released to the intended victim.

RAT servers can be customized using binders, which are RAT-provided configuration packages, before installation. Encryption algorithms, autostart methods, default TCP/UDP port definitions, and initial login passwords are all part of this customization [12]. Any RAT attach operation has three steps:

- The attacker creates an executable file. Often referred to as server.exe, this is the trojan's server component.
- The attacker associates this server.exe with any legitimate file, such as a picture or music. The attacker sends this file to the victim, who is meant to open it or click on it. The victim is then asked to run server.exe.
- When the victim runs the server portion, a port on the victim's machine opens and attempts to establish a connection with the attacker. Through the control panel, the attacker can thus remotely manipulate the victim's computer from anywhere through the Internet.

RATs are still one of the most infected malware, as shown in Figure 1 [27].

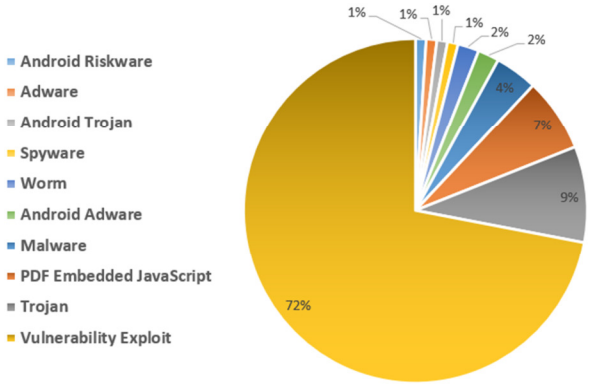

Fig. 1. Malware distribution.

III. METHODS OF MALWARE ANALYSIS AND DETECTION

A. Analysis Methods

Malware can be analyzed either statically or dynamically, and both approaches have their use.

1) Static Analysis

The field of automatically deducing details about computer programs without actually running them is known as static analysis [28]. Various techniques and tools are used to differentiate a file from a non-harmful one. Static analysis collects technological indicators and generates basic signatures by providing details about its functionality. File names, file types, file sizes, and MD5 checksums or hashes are examples of technical indications [29].

2) Dynamic Analysis

Dynamic analysis is carried out by actually executing the code and observing its functionality and behavior [26]. Because it examines the program while it is running, it assists in detecting code obfuscations, polymorphic malware, and unpacking malware, which are some of the static analysis's limitations. This is one way to examine the actual behavior of a

program. However, there are some disadvantages. The primary one is dormant code: Dynamic analysis is typically limited to tracking a single execution path and is unable to cover the entirety of the code. A poorly isolated or restricted analysis environment will make the third-party system dangerous. Malware may also stop executing or behave differently if it recognizes that it is running in a controlled analysis environment [30].

B. Detection Methods

Detection methods can be classified as signature-based and behavior-based, depending on the analysis approach. In addition, malware detection methods are classified into host-based and network-based detection methods, according to where the system is introduced.

1) Signature-based vs Behavior-based Detection System

Signature-based detection is one of the conventional techniques for detecting malware. This technique evaluates malicious network communications using pre-established signatures [29]. Static analysis serves as the foundation for most of these techniques. A database is created that contains signatures that are taken from known malware attacks. An antivirus scanner is an application that uses signature-based detection. On the other hand, in the behavior-based malware detection method, malware can be detected by modeling peculiar behaviors that are different from the normal state [31]. These techniques gather the behavior of different malware, as some different malware behaviors are comparable. Thus, a single action can reveal multiple malware. Because they function similarly in the system and use the same resources, this kind of detection technique aids in the discovery of new malware variants [32]. Table I summarizes the differences between signature- and behavior-based malware detection.

TABLE I. DIFFERENCES BETWEEN SIGNATURE AND BEHAVIOR-BASED MALWARE DETECTION

Signature-based malware detection	Behavior-based malware detection
Need of huge database with daily updates	Database is smaller
Can detect known attacks efficiently but cannot detect new unknown malware	Can detect new unknown malware
Low false positive rate	High false positive rate
Fewer resources are required	Need more resources like CPU time and memory in monitoring.
Static-based malware detection	Dynamic-based malware detection

2) Host-based vs Network-based Detection System

A Host-based Intrusion Detection System (HIDS) gathers data on the activities of a single system or host. A server or an individual computer is called a host. Host-based agents, also known as sensors, are installed on a machine that may be the target of an attack. HIDS operates continuously, keeping an eye on who has access to the system and what applications are used, notifying users of any unexpected activity. It monitors system logs, application logs, user event logs, file integrity, rootkit detection, policy enforcement, and other intrusions into the system. These logs are used to construct a baseline. When fresh activity starts, HIDS compares it to the baseline and raises an alarm if any logs are discovered to be outside normal use. If any illegal behavior is detected, HIDS will either stop it, notify the user, or take another action following the settings made by the system administrator [33, 34].

A Network-based Intrusion Detection System (NIDS) is responsible for the surveillance of both incoming and outgoing network traffic within an organization's network infrastructure, with the primary objective of identifying and mitigating potentially harmful behaviors. The NIDS is strategically deployed within a certain location in the network infrastructure, through which both incoming and outgoing network traffic is traversed. The system examines each packet that traverses the network and compares the data with its repository of signatures. If new entries are detected in the database, it immediately sends alarms to the designated management console [35]. Table II summarizes the differences between host- and network-based malware detection.

TABLE II. DIFFERENCES BETWEEN HOST AND NETWORK-BASED MALWARE DETECTION

Host	Network
Should be installed in each host	It can be placed in any central device
Software based	Hardware-based
Bandwidth independent	Bandwidth dependent
Does not examine the packet header	Examines the packet header and the entire packet.

To detect and analyze potentially malicious activities, network traffic was monitored and analyzed in real-time using the Wireshark software tool. Sysinternals tools were used to investigate active processes, system startup items, and registry keys for any irregularities. Moreover, VirusTotal was used to submit dubious files for analysis and gain a supplementary understanding of their behavior and reputation. A thorough examination of the network and system was carried out by integrating these tools, facilitating the detection of possible RAT attacks.

IV. IMPLEMENTING THE PROPOSED ENVIRONMENT

The virtual lab environment was created using VMware. The lab consists of three virtual machines running under Windows 10 64-bit. The virtual machines are connected to the LAN with static IPs, as shown in Table III.

TABLE III. VIRTUAL MACHINE SETTINGS

No.	VM Name	Role	IP Address
1	RAT-Client	Attacker	192.168.0.122
2	RAT-Server1	Victim	192.168.0.133
3	RAT-Server2	Victim	192.168.0.134

The malicious script is built to infect victims. The following steps were taken to build a server using Mega RAT 1.5 Beta.

Step 1: Run Mega RAT 1.5 Beta and click on Create Server, as shown in Figure 2.

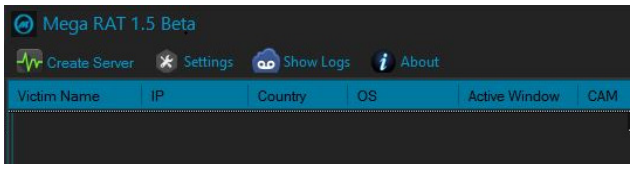

Fig. 2. Building the RAT server.

Step 2: Write the IP address of RAT client and the port number to listen on the victim's PC (port 2020), as shown in Figure 3.

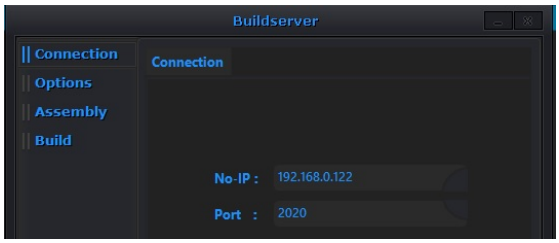

Fig. 3. Selecting IP and port.

Step 3: Write a name for the victim to distinguish it when he logs in and select an attractive name for the malicious file (in this case, the name COVID-19Live was used), as shown in Figure 4.

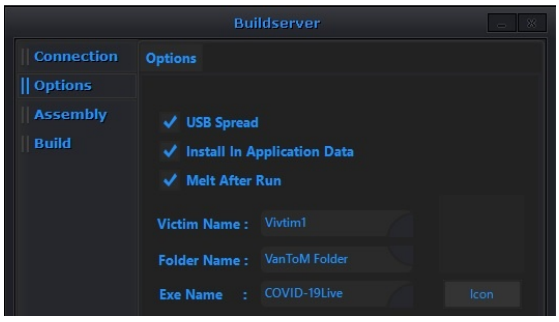

Fig. 4. Selecting the name for the RAT server.

Step 4: Select Generate All to activate the server, as shown in Figure 5.

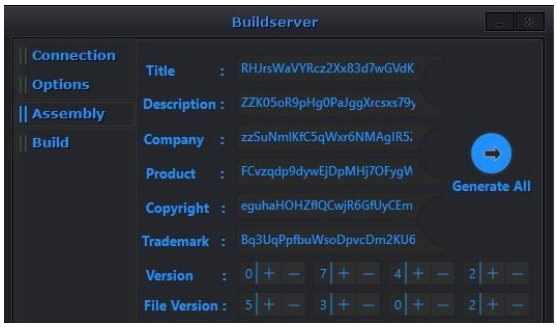

Fig. 5. Activating the RAT server.

Step 5: Choose an executable file extension for the server file (the .exe extension was used), as shown in Figure 6.

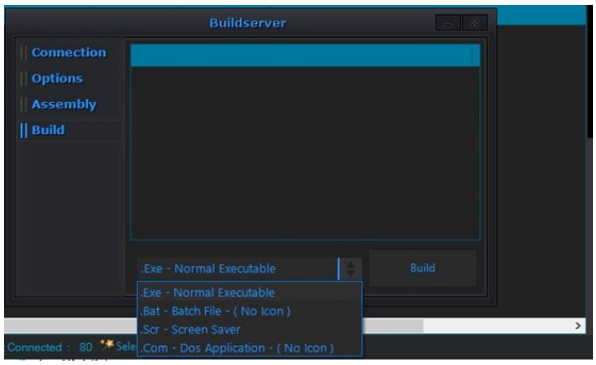

Fig. 6. Selecting RAT server extension.

Step 6: Choose Build and save the file in any location. Then, the server is finally built, as shown in Figure 7.

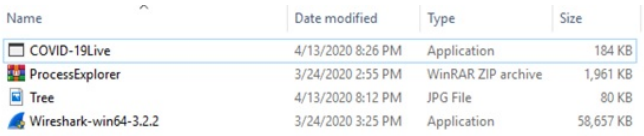

Fig. 7. The RAT server file.

Step 7: Copy the RAT file to a LAN-shared folder or USB and give it to the victim. A sound notification informs the attacker when the victim runs the server file. The attacker can control the victim's PC and do many things, as shown in Figures 8, 9, and 10.

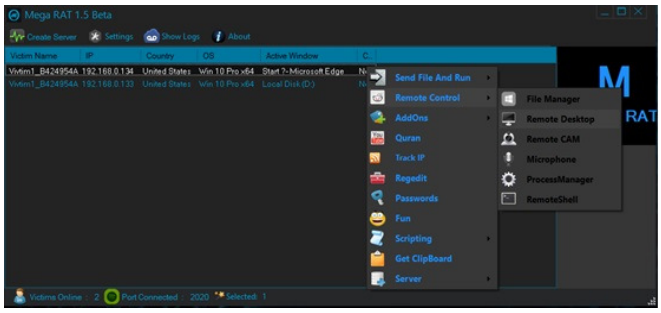

Fig. 8. Activation of remote desktop.

Fig. 9. The desktop of the victim.

Fig. 10. Some fun commands.

As soon as the target victim opens the generated RAT, the executor gains complete access to the target machine, allowing him to monitor the target's activities.

V. RESULTS AND DISCUSSION

There is no standard approach for analyzing malware. First, a static analysis was performed without running the malware, and then a dynamic analysis was applied, in which its activities were examined by running the malware in the lab environment. The static analysis of COVID-19Live.exe was tested by the www.virustotal.com website, which consists of databases of the most famous antivirus companies. Figure 11 shows that the subject file was distinguished as malware in 57 of 70 antiviruses.

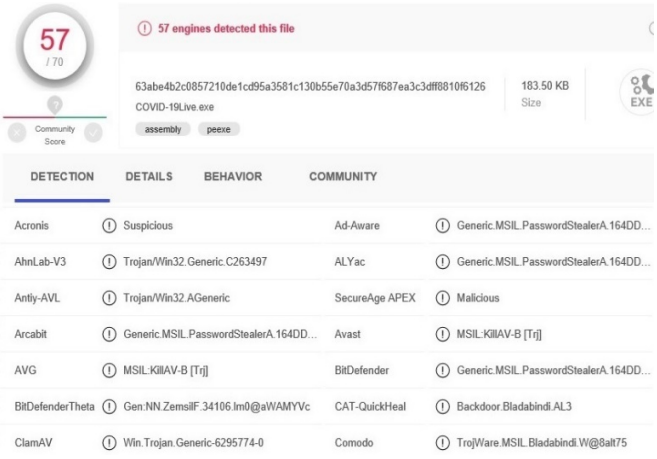

Fig. 11. Virustotal test results.

An analysis of Mega RAT 1.5 Beta's characteristic behavior was performed through Wireshark and Process Explorer. Figure 12 shows the COVID-19Live.exe traffic captured by Wireshark on the victim's machine.

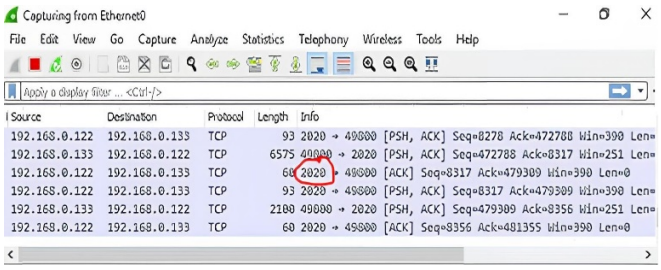

Fig. 12. Wireshark screenshot.

The Process Explorer (SysInternals) was used to monitor the system process, as shown in Figure 13. It is similar to the Windows Task Manager, with more freedom in getting information. It shows all the current processes running in the system, along with their children, descriptions, process IDs, and many other useful information. COVID-19KLive was colored purple, indicating that this file is packed (encrypted) and is more likely to be malware. In addition, the SysInternals virus check tool was used, showing that this file is unknown.

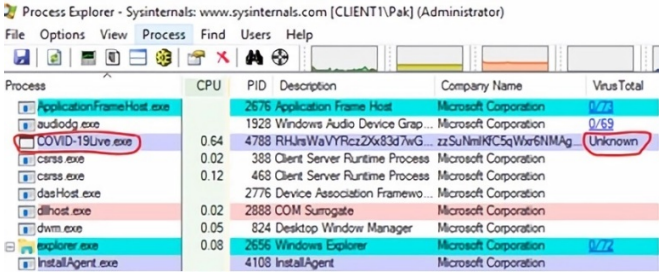

Fig. 13. Process Explorer screenshot.

Table IV shows a comparison between the research methodology in this and previous studies.

TABLE IV. COMPARISON BETWEEN DIFFERENT METHODOLOGIES

Ref.	Methodology	Detection/Analysis Methods
[7]	Two types of trojan attacks (RAT and HTTP) were used and experimented with in a network environment	Wireshark was used to analyze suspicious packets
[8]	An experimental testbed was set up with two laptops connected via a router. One laptop acted as the attacker, and the other as the victim	The presence of malware was detected using VirusTotal MalwareBytes and Avast antivirus.
[9]	Created a RAT server and infected a victim's system using Beast 2.06 RAT	Spyware detection tools, signature-based detection, and TCP/UDP port monitoring
[10]	Exploitation approach using the Ghost framework and Android Debug Bridge (ADB)	Employed an exploitation approach to test for vulnerabilities on Android devices
[11]	Installed Kali Linux and FTK Imager, designed and generated RAT virus and network topology	Disk forensics and memory forensics using FTK Imager software
This study	Created a RAT server and infected a victim's system using Beast Mega RAT 1.5	The presence of malware was detected using VirusTotal, Wireshark, and Sysinternals

VI. CONCLUSIONS

This study implemented a virtual experimentation environment for a real RAT attack using Mega RAT 1.5 Beta. These experiments can provide an understanding of user awareness of RAT detection to prevent the loss of personal information. In addition, a variety of static and dynamic analyses were performed using Wireshark and Sysinternals to investigate the malware. It was determined that the static tools used were more reliable than the dynamic analysis tools. However, their use requires access to an international database that is updated daily. In addition, the static analysis method does not require any prior knowledge about malicious software. Future studies could experiment with RATs transmitted over the Internet employing different software, testing other malware detection methods, and investigating nested RATs.

REFERENCES

[1] K. S. Yin, "Network Behavioral Analysis for Detection of Remote Access Trojans," Ph.D. dissertation, University of Computer Studies, Yangon, Myanmar, 2019.

[2] V. Valeros and S. Garcia, "Growth and Commoditization of Remote Access Trojans," in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, Genoa, Italy, Sep. 2020, pp. 454–462, https://doi.org/10.1109/EuroSPW51379.2020.00067.

[3] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware Trojans: Lessons Learned after One Decade of Research," *ACM Transactions on Design Automation of Electronic Systems*, vol. 22, no. 1, Feb. 2016, https://doi.org/10.1145/2906147.

[4] I. Androulidakis and G. Kandus, "Mobile Phone Brand Categorization vs. Users' Security Practices," *Engineering, Technology & Applied Science Research*, vol. 1, no. 2, pp. 30–35, Apr. 2011, https://doi.org/10.48084/etasr.19.

[5] B. N. Bukke, K. Manjunathachari, and S. Sabbavarapu, "Implementation of a Finite Impulse Response Filter using PUFs to Avoid Trojans," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12151–12157, Dec. 2023, https://doi.org/10.48084/etasr.6133.

[6] A. Alshammari, "A Novel Security Framework to Mitigate and Avoid Unexpected Security Threats in Saudi Arabia," *Engineering, Technology & Applied Science Research*, vol. 13, no. 4, pp. 11445–11450, Aug. 2023, https://doi.org/10.48084/etasr.6091.

[7] M. A. Hashim *et al.*, "Digital Forensic Investigation of Trojan Attacks in Network using Wireshark, FTK Imager and Volatility," *Journal of Computing Research and Innovation*, vol. 2, no. 2, pp. 60–65, Jun. 2017.

[8] A. M. Taib and N. N. K. A. Azman, "Experimental Analysis of Trojan Horse and Worm Attacks in Windows Environment," *Journal of Advanced Research in Computing and Applications*, vol. 13, no. 1, pp. 1–9, 2018.

[9] S. Mirdul, "A Study on RAT (Remote Access Trojan)," *Academic Journal of Forensic Sciences*, 2019.

[10] D. Aprilliansyah, I. Riadi, and Sunardi, "Analysis of Remote Access Trojan Attack using Android Debug Bridge," *IJID (International Journal on Informatics for Development)*, vol. 10, no. 2, pp. 102–111, 2021, https://doi.org/10.14421/ijid.2021.2839.

[11] A. H. Hendrawan, R. Kurniawan, A. J. Aprian, D. Primasari, and M. Subchan, "Enhancing Cybersecurity Through Live Forensic Investigation of Remote Access Trojan Attacks using FTK Imager Software.," *International Journal of Safety & Security Engineering*, vol. 14, no. 1, 2024.

[12] M. N. Kondalwar and C. J. Shelke, "Remote Administrative Trojan/Tool (RAT)," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 3, pp. 482–487, Mar. 2014.

[13] L. Fu, "Design of Hidden Communication Remote Monitoring Based on C / C MFC," in *2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, Hohhot, China, Oct. 2019, pp. 589–5892, https://doi.org/10.1109/ICMCCE48743.2019.00135.

[14] I. Kennedy, A. Bandara, and B. Price, "Towards Increasing Trust In Expert Evidence Derived From Malware Forensic Tools." arXiv, Oct. 14, 2020, https://doi.org/10.48550/arXiv.2010.07188.

[15] C. Wuest, "Advanced communication techniques of remote access trojan horses on Windows operating system GSEC Practical v1. 4b (option 1)," 2004.

[16] A. Spalka, A. B. Cremers, and H. Langweg, "The fairy tale of what you see is what you sign - trojan horse attacks on software for digital signatures," in *Proceedings of the IFIP WG*, 2001, vol. 9, no. 11.7, pp. 75–86.

[17] Q. A. Al-Gburi and M. A. Mohd Ariff, "Dynamic Security Assessment for Power System Under Cyber-Attack," *Journal of Electrical Engineering & Technology*, vol. 14, no. 2, pp. 549–559, Mar. 2019, https://doi.org/10.1007/s42835-019-00084-2.

[18] S. Gadhiya, K. Bhavsar, and P. D. Student, "Techniques for malware analysis," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 4, 2013.

[19] M. Mohd Saudi, A. M. Abuzaid, B. M. Taib, and Z. H. Abdullah, "Designing a New Model for Trojan Horse Detection Using Sequential Minimal Optimization," in *Advanced Computer and Communication Engineering Technology*, 2015, pp. 739–746, https://doi.org/10.1007/978-3-319-07674-4_69.

[20] C. Jin, X. Y. Wang, and H. Y. Tan, "Dynamic Attack Tree and Its Applications on Trojan Horse Detection," in *2010 Second International Conference on Multimedia and Information Technology*, Kaifeng, China, Apr. 2010, vol. 1, pp. 56–59, https://doi.org/10.1109/MMIT.2010.12.

[21] Y. Kang, X. Yu, W. Meng, and Y. Liu, "BlockRAT: An Enhanced Remote Access Trojan Framework via Blockchain," in *Science of Cyber Security*, Matsue, Japan, Aug. 2022, pp. 21–35, https://doi.org/10.1007/978-3-031-17551-0_2.

[22] D. Jiang and K. Omote, "An Approach to Detect Remote Access Trojan in the Early Stage of Communication," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*, Gwangju, Korea (South), Mar. 2015, pp. 706–713, https://doi.org/10.1109/AINA.2015.257.

[23] M. B. Johansen, "Development of a customized remote access trojan (RAT) for educational purposes within the field of malware analysis," M.S. Thesis, Norwegian University of Science and Technology, 2022.

[24] D. Adachi and K. Omote, "A Host-Based Detection Method of Remote Access Trojan in the Early Stage," in *Information Security Practice and Experience*, Zhangjiajie, China, Nov. 2016, pp. 110–121, https://doi.org/10.1007/978-3-319-49151-6_8.

[25] G. Karantzas and C. Patsakis, "An Empirical Assessment of Endpoint Detection and Response Systems against Advanced Persistent Threats Attack Vectors," *Journal of Cybersecurity and Privacy*, vol. 1, no. 3, pp. 387–421, Sep. 2021, https://doi.org/10.3390/jcp1030021.

[26] M. Marchetti, F. Pierazzi, M. Colajanni, and A. Guido, "Analysis of high volumes of network traffic for Advanced Persistent Threat detection," *Computer Networks*, vol. 109, pp. 127–141, Nov. 2016, https://doi.org/10.1016/j.comnet.2016.05.018.

[27] N. Nissim *et al.*, "Scholarly Digital Libraries as a Platform for Malware Distribution," in *A Systems Approach to Cyber Security*, IOS Press, 2017, pp. 107–128.

[28] B. Dang, A. Gazet, and E. Bachaalany, Practical Reverse Engineering: x86, x64, ARM, Windows Kernel, Reversing Tools, and Obfuscation. John Wiley & Sons, 2014.

[29] A. Moser, C. Kruegel, and E. Kirda, "Limits of Static Analysis for Malware Detection," in *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*, Dec. 2007, pp. 421–430, https://doi.org/10.1109/ACSAC.2007.21.

[30] L. Caviglione *et al.*, "Tight Arms Race: Overview of Current Malware Threats and Trends in Their Detection," *IEEE Access*, vol. 9, pp. 5371–5396, 2021, https://doi.org/10.1109/ACCESS.2020.3048319.

[31] M. Oya and K. Omote, "Early Detection of Remote Access Trojan by Software Network Behavior," in *Information Security and Cryptology*,

Fuzhou, China, Dec. 2018, pp. 658–671, https://doi.org/10.1007/978-3-030-14234-6_37.

[32] M. S. Nawaz, P. Fournier-Viger, M. Z. Nawaz, G. Chen, and Y. Wu, "MalSPM: Metamorphic malware behavior analysis and classification using sequential pattern mining," *Computers & Security*, vol. 118, Jul. 2022, Art. no. 102741, https://doi.org/10.1016/j.cose.2022.102741.

[33] U. H. Rao and U. Nayak, The InfoSec Handbook: An Introduction to Information Security. Berkeley, CA, USA: Apress, 2014.

[34] A. S. K. Pathan, The State of the Art in Intrusion Prevention and Detection. Auerbach Publications, 2014.

[35] M. Mwita, J. Mbelwa, J. Agbinya, and A. E. Sam, "The Effect of Hyperparameter Optimization on the Estimation of Performance Metrics in Network Traffic Prediction using the Gradient Boosting Machine Model," *Engineering, Technology & Applied Science Research*, vol. 13, no. 3, pp. 10714–10720, Jun. 2023, https://doi.org/10.48084/etasr.5548.