Dissecting Digital Dangers: A MITRE-Aligned Analysis of Two Prominent Malware Instances

Subhiksha Suresh Rao, Richa Vivek Savant, Sahil Mishra, Sreevidya B., Rajesh M.*
Department of Computer Science and Engineering,
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
*Corresponding Author: m_rajesh@blr.amrita.edu

***Abstract*—The increasing complexity of malware requires a more thorough comprehension of how they operate. Using the MITRE ATT&CK framework, this project delves into the TrickBot and AsyncRAT malwares, offering a thorough examination of their tactics, techniques, and procedures (TTPs) across the cyber kill chain. This study uncovers trends in how these malicious software gain entry, carry out tasks, remain active, and extract information from a network by correlating TTPs with ATT&CK methods. Key analyses indicate that TrickBot utilizes elevated entropy values to avoid being detected, incorporates modular structures to enhance its capabilities, and utilizes spear-phishing and network spreading to achieve broad infection. AsyncRAT is found to employ sophisticated anti-analysis methods, reflective code loading, and achieves persistence by altering the system registry while avoiding sandbox environments. Additionally, it establishes command and control (C2) connections to maintain long-term access over compromised systems. The observations uncovered in this research emphasize the importance of having strong cybersecurity policies and comprehensive threat intelligence to combat changing threats effectively. Understanding the complex behaviours and evasion methods of these malwares is crucial for creating specific defences and improving security overall.**

***Index Terms*—MITRE ATT&CK Framework, TrickBot, AsyncRAT, TTPs, Cyber Kill Chain, Spear-phishing, Persistence, Command and Control (C2)**

I. Introduction

In the ever-shifting landscape of cyber threats, two adversaries stand out: AsyncRAT and TrickBot. This paper delves into their malicious machinations. Two modern malware samples of AsyncRAT and Trickbot, respectively, are analysed thoroughly to gain a deep understanding of their functionalities, behaviours, and impacts. The MITRE ATT&CK framework provides in-depth information about tactics and procedures of malicious software, shedding light on key steps in the cyber kill chain.

AsyncRAT is a .NET-based Remote Access Trojan (RAT), a symbol of the growing cyber threat landscape. It represents the sophistication and adaptability of modern malware, which uses antivirus evasion techniques to infiltrate systems, create stability, and allow for remote access. An in-depth analysis explains AsyncRAT's complex network communication protocols, stealth execution mechanisms, and strategies to maintain stability in compromised environments.

In contrast, TrickBot is a popular credential stealer known for its wide range of features. Research at TrickBot includes various attack vectors, including spear phishing campaigns, exploits, and network propagation techniques. TrickBot uses an advanced command and control infrastructure to coordinate anti-virus evasion strategies and financial cybercrime operations.

This study delves into the importance of understanding malware behaviour within the MITRE ATT&CK framework. Examples of AsyncRAT and TrickBot TTPs map to specific ATT&CK technologies to reveal initial access, execution, persistence, escalation, and data exfiltration strategies. Moreover, the complicated interactions between these malware samples and defensive countermeasures are examined, showcasing the importance of proactive threat intelligence and a robust cybersecurity strategy. Ultimately, this study aims to provide cyber security professionals with actionable insights to counter modern malware threats by revealing the inner workings of AsyncRAT and TrickBot in the context of cyber killing channels.

This work proposes a detailed analysis of these malware samples through static and dynamic analysis techniques. Static analysis involves using tools such as Detect It Easy, VirusTotal, and HashCalc to identify the malware structure, entropy, and hash values. Dynamic analysis in which Process Monitor, Procmon is used to track the events that the malware would perform when it is executed. The code analysis includes the tool dnSpy. The result will answer how the malware evades antivirus detection and uses persistence mechanisms. TrickBot and AsyncRAT TTPs are mapped to the MITRE ATT&CK framework to establish the patterns actionable insight cybersecurity professionals will use to develop effective countermeasures.

II. Literature Survey

Zhang et al. [1] developed a malware detection model using API2Vec and Balts for API call features. It outperforms existing methods by 3-5% in detection accuracy. However, the BiLSTM module's impact is marginal, indicating room for improvement.

Al-sofyani et al. [2] provide an overview of various malware forensics techniques and tools. They emphasize the need for malware analysis and detection to protect economies and

corporate assets. The paper discusses diverse analysis and detection techniques and aims to equip IT workers with skills to mitigate malware threats.

Nair et al. [3] highlight the critical threat of malicious code to computer systems, necessitating robust malware analysis methods. They focus on static analysis as an initial defence and discuss persistent challenges in detecting evasive malware. The authors point out the limitations of current detection approaches and the need for rapid learning tactics for zero-day attacks. They suggest leveraging AI and machine learning advancements to combat evolving malware threats.

Thakur's study [4] involves an in-depth examination of Trickbot malware's post-exploitation effects. The paper analyzes its targeting techniques, including industry sectors and geographic regions, highlighting its dynamic victim selection. The research also explores Trickbot's use of wild-carded URIs to expand targets and evade detection. The findings emphasize its versatility and advanced penetration capabilities beyond the traditional banking sector.

Thakur [5] analyzes the TrickBot malware variant's execution sequence and provides information on its injection, configuration loading, evasion, and execution processes. It describes how the malware gets around local anti-malware programs, including how to disable and remove the Windows Defender service. The study improves comprehension of the malware's evasion strategies, execution process, and insertion approaches by analyzing its behaviour and mechanics. The results demonstrate how cleverly TrickBot is designed, emphasizing its capacity to elude anti-malware defences and carry out hostile operations in stealth.

Sikander's paper [6] examines AsyncRAT, a Remote Access Trojan linked to the hacker group Blind Eagle. The malware uses spear phishing, software flaws, and social engineering, as well as process manipulation and AES-256 encryption to bypass security. The paper argues for behavioural detection in cybersecurity and notes the challenge of detecting polymorphic malware like AsyncRAT.

Divya et al. [7] explore the use of command and control servers and Domain Generation Algorithms (DGAs) by attackers. They propose a real-time detection system for word-based DGA domain names using techniques like graph theory and feature extraction. The system's performance evaluation shows promising results, with Random Forest having the highest accuracy. The paper suggests incorporating additional features to improve the system's detection capabilities.

Murali et al. [8] discuss the merger of bio-inspired computing and malware research. They highlight the benefits, like enhancing malware databases and mitigating zero-day attacks, and challenges, such as defining "maliciousness". The paper calls for more research in this field, emphasizing its potential to improve cyber safety.

Bhatia et al. [9] conducted a comprehensive review of malware analysis literature, focusing on evasion techniques. They detailed a process for dynamic analysis, useful for researchers and practitioners. Tools like Wireshark and ProcDot were used to study malware behaviour, providing insights for developing defence strategies and improving system security.

Sinha et al. [10] combine static and dynamic analysis for malware in a virtual environment, emphasizing the need for adaptable cybersecurity against evolving threats. Their research reviews various effective analysis techniques such as audits, PE file analysis, machine learning models, and sandboxing. Sandboxing, in particular, provides a potential solution for identifying domain-related threats.

Maher et al. [11] thoroughly explored malware examination techniques via reverse engineering. They highlight the intertwining of malware and anti-detection mechanisms and the importance of static and dynamic analysis against cyber threats. They advocate for malware analysis frameworks that use reverse engineering to improve detection accuracy and efficiency.

Roshini et al. [12] developed a new method to observe malware behaviour and its impact on co-occurrence signatures and regression modelling. The method aims to accurately measure the severity of these signatures, outperforming existing score systems. The study demonstrated the method's ability to distinguish signatures based on their correlation and identified areas for improvement, such as scoring precision for low signature samples.

Balasubramanian et al. [13] used memory analysis for malware detection with the CIC-Malmem-2022 dataset on Google Colab. They prioritized pre-processing and feature engineering and used Decision Trees and Random Forest algorithms, achieving high detection accuracy. The study underscores the value of memory analysis and machine learning optimization in tackling cyber threats.

Mithun et al. [14] developed a framework that combines static and dynamic analysis to detect stalkerware applications effectively. It surpassed existing antivirus software with a 95.9% accuracy rate. The paper suggests further refining dynamic analysis for precise detection.

Paul et al. [15] discuss static and dynamic malware analysis, highlighting their importance in examining files and observing behaviour. They stress the need for malware classification and propose extracting behaviour artefacts' features to develop an efficient malware detection system. The paper concludes with a proposal to extract features from behaviour artefacts to develop a highly efficient malware detection and classification system in future works.

III. Methodology

A. Non-Technical Analysis of TrickBot

1) Origin of TrickBot: TrickBot has its roots as a malware that steals banking credentials, but now it is seen as an enterprise of modular viruses. First identified in 2016, TrickBot was designed by an advanced group of cyber criminals who continue to maintain and upgrade it. The Trojan has grown into multi-stage malware with many capabilities for different types of illegal cyber activity. The main aim behind TrickBot is to steal sensitive data or login details; however, there are

also other modules known for turning it into a full-fledged malware service. For example, one such module delivered Cobalt Strike which eventually resulted in Ryuk and Conti ransomwares being activated.

2) Entities Ensnared in the TrickBot Malware's Crosshairs: TrickBot initially emerged as a major player in the banking malware game in 2016. New targets were constantly added to its lineup, which started with banks in various regions before expanding to include new financial institutions every month.

At its height, the profile featured American, European, Australian and Asian banks – primarily Indian ones like ICICI and HDFC. The attackers likely realized at some point that they should concentrate on Western nations for greater profits. Most initial infections involved top-tier and upper second-tier banks in Australia. There is only one Australian bank left on TrickBot's target list – CBA (which was also among the first financial institutions ever targeted by the malware when it was dropped).

B. Static Analysis of TrickBot

1) Malware Bazaar Lookup: As seen in Fig. 1 An 8-year-long and ongoing search of TrickBot tags on Malware Bazaar came up with many examples. A fairly new sample from the year 2021 was downloaded safely in a Windows 10 sandboxed environment.

An isolated malware sandbox is a safe place to execute potentially malicious software away from the main operating system so as to observe its behaviour. It is a containment method which is created with the main purpose of preventing any possible malware from infecting the network or system and giving analysts an opportunity to study its functionality and aim without risk.

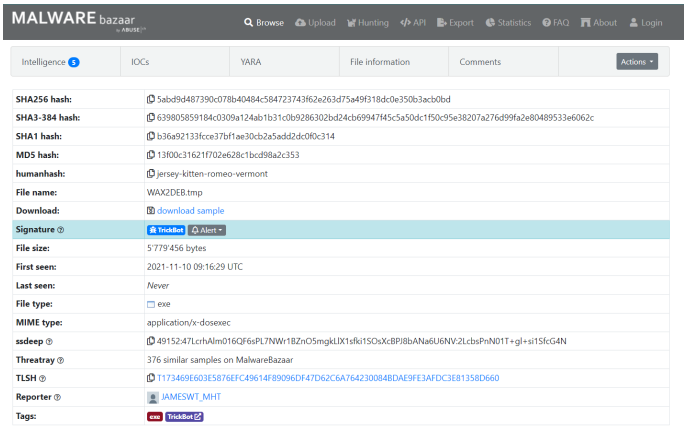

Fig. 1. Malware Bazaar Lookup of TrickBot

2) Reconnaissance using VirusTotal: The MD5 hash is a unique identifier created from the sample and used to search the VirusTotal platform. This platform brings together different antivirus software and online scan engines to detect viruses that might have escaped detection by the user's own antivirus or identify false positives. Among 70 antiviruses, 29 identified the sample as malicious.

This outcome demonstrates that a wide variety of anti-virus solutions tested agreed on treating this specimen as malware. Nevertheless, it should be mentioned that not all anti-virus programs recognized it as so. Such inconsistency may stem from diverse detection algorithms and heuristics employed by various anti-virus products. In general terms, the VT results strongly suggest the presence of malice in the sample thereby underpinning the need for deep examination and appropriate defensive actions.

3) Initial Analysis Using Detect It Easy Software: The 'Detect It Easy' Software was used for the first analysis of the TrickBot malware sample as observed in Fig. 2, and many important discoveries were made that shed light on the virus's properties. The file type was found to be PE32 (32-bit portable executable). This format is commonly used for Windows executable files.

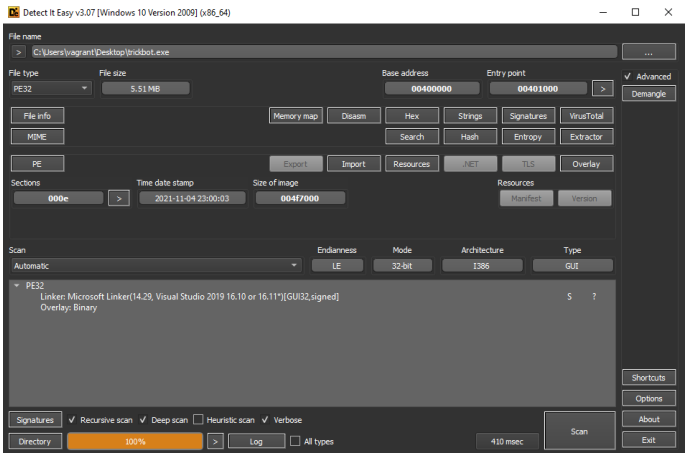

Fig. 2. Detect It Easy Findings on the TrickBot sample

The size of the file was recorded as 5.51 MB which is fairly large for malware samples suggesting there may be many complex or different functions contained within it. The time date stamp shows that the file was last modified on 2021-11-04 at 23:00:03. The linker was identified as Microsoft Linker (Visual Studio 2019), meaning that it could have been compiled in an environment with these tools. Moreover, the overlay is binary. This is a common technique employed by malicious software creators to conceal extra information inside a file; such data can range from simple configuration details to additional dangerous payloads – thus making it more potent or letting them update/control it when already implanted into some system.

The following notable findings are recorded:

- File type: PE32 (32-bit portable executable)
- File size: 5.51 MB
- Time date stamp: 2021-11-04 23:00:03
- Linker: Microsoft Linker(14.29, Visual Studio 2019 16.10 or 16.11*)[GUI32,signed]
- Overlay: Binary

All in all, these findings give a clear idea about the TrickBot malware sample's complexity levels.

4) Calculating Hash Values: To find out the hash values of the malware sample, HashCalc is used, as shown in Fig. 3. Among the generated hash values were MD5, MD4, SHA1 and SHA256. These serve as fingerprints for a file's digital content which are unique in nature. The field of digital forensics and incident response strongly relies on them as they help researchers identify, categorize and even trace malicious software.

For example, the cryptographic hash function known as MD5 creates a 16-byte (128-bit) hash value. It is often used to verify data integrity. By generating these hash values and comparing them, experts can track the malware across different systems and networks. This helps in detecting any changes that may occur and allows them to adjust their defense strategies accordingly.

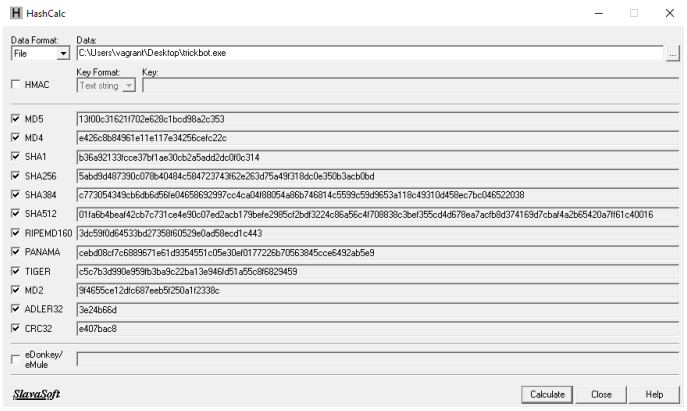

Fig. 3. HashCalc Findings on the TrickBot sample

5) Analyzing Entropy of the Sections: The entropy value of a file indicates its randomness or disorder. When analyzing malware, a high entropy value typically implies the use of encryption or packing techniques for hiding the code. In this instance, as observed in Fig. 4 the total entropy value of the TrickBot sample was found to be 6.39512; which is quite high (the maximum entropy value being 8), indicating that there may exist some level of intricacy in the obfuscation employed by the file.

It should be noted that although high values can indicate packing, they do not prove it. Notably, four sections of this file were packed. This means that only parts were hidden while others remained visible. Malware authors might have done this as a way to selectively conceal some portions of their codes while leaving other parts open for anyone to see through such as researchers or analysts.

6) Extracting Strings from the Binary: In order to extract strings, the binary data in the file must be analyzed using Detect It Easy to discover printable character sequences. Potential command and control servers, file paths, registry keys, and other important signs of compromise can all be found using this technique. Hence, some strings might be encoded or encrypted if the malware employs obfuscation techniques, which would make them harder to decipher.

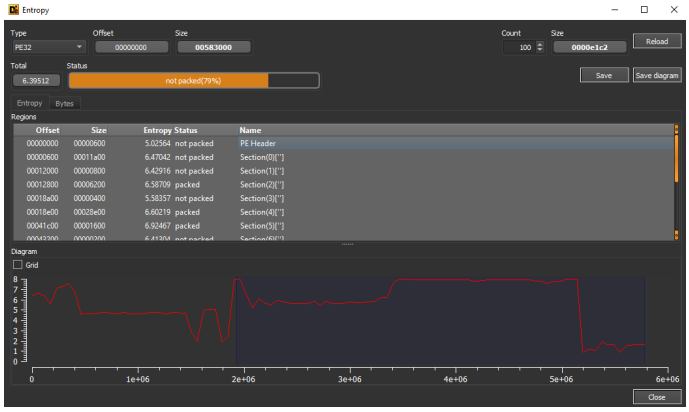

Fig. 4. Detect It Easy Entropy Statistic Findings of the TrickBot sample

C. Non-Technical analysis of AsyncRAT

1) Tracing the Roots of AsyncRAT: AsyncRAT is a C#-written remote access trojan RAT with very mature functions of keylogging and remote desktop management, hence posing a grave threat to victims. It adopts an event-based, asynchronous operating mode that allows cybercriminals to manage target systems to execute diverse malicious activities. It is attributed to the South American group Blind Eagle, otherwise known as APT-C-36. This group targets high-value entities from the Colombian government, banking, oil, and specialized industries.

2) Inside the mind of an AsyncRAT sample: Blind Eagle APT's phishing campaign starts with a deceptive email featuring a password-protected PDF and a Spanish subject line, attempting to entice users with an urgent tax document. Upon opening, users find a URL mimicking the Directorate of National Taxes and Customs' official site. This illegitimate link redirects to another site controlled by the attackers, stealthily extracting a payload from a public Discord server. The sophistication of this attack underscores the importance of vigilance and robust cybersecurity in combating phishing.

D. Static Analysis of AsyncRAT

1) A preliminary analysis using Detect It Easy:

- Filetype: PE32
- Library: .NET(v4.0.30319)[-]
- Compiler: [VB.NET](http://vb.net/)(-)[-]
- Linker: Microsoft Linker(8.0)[GUI32]
- MD5 hash: c0b9838ff7d2ddecbfe296eae947e5d6
- SHA1: 76af794b85e4a4ba75c5703df1207b7a6798bf2e
- File size: 45.00 KiB
- Timestamp: 2020-05-10 05:24:51

2) Entropy levels of each section: Entropy values can be considered as indicators of file packing. Typically, packed files exhibit higher entropy values since packing involves compression and encryption to disguise the file's content by reducing its size. The sample has an entropy value of 5.45814

which is relatively low. Therefore, Detect It Easy (DIE) infers that the sample is unpacked.

3) Virus Total lookup of the MD5 hash: The MD5 hash is a unique identifier for the file and is used to cross-reference AsyncRAT samples on the VirusTotal website. This example has been flagged as malicious by 57 of the 68 security providers and sandboxes available on the site. This indicates that the vast majority of cybersecurity tools detect the probe as a security threat. Such high detection rates highlight the need for strong and updated antivirus software to protect against threats like AsyncRAT.

4) TTP (Tactics, techniques and procedures) analysis using Capa: The ATT&CK Tactic and Technique analysis conducted using Capa reveals the following about the AsyncRAT malware:

- Under the COLLECTION tactic, the malware gathers and stores data using specific libraries.
- Several techniques under the DEFENSE EVASION tactic show that the malware prevents detection by disguising files, altering the system registry, and evading virtual or sandbox environments.
- The DISCOVERY tactic involves the malware seeking extensive information about the infected system, such as account details, files and directories, active processes, registry entries, installed software, overall system information, and user details.
- The EXECUTION tactic indicates that the malware utilizes Windows Management Instrumentation for carrying out its operations.
- Under the PERSISTENCE tactic, the malware ensures its continuous presence on the infected system, likely by scheduling tasks or jobs.

E. Code Analysis of The AsyncRAT Sample:

An elementary examination of the .NET code was conducted with dnSpy. dnSpy, a robust .NET debugger and assembly editor, enables security researchers to analyze the composition and features of .NET malware.

- A for loop runs four times and employs the Thread.Sleep(1000) instruction, in order to evade AV tests.
- The sample then executes anti-analysis checks using the Anti_Analysis.RunAntiAnalysis() command for defence evasion.
- Using ProcessCritical.Set(), it positions itself as a critical process in order to prevent detection and removal by AV/EDRs. This is carried out in case a crucial operation is ended, which could result in a Blue Screen of Death (BSOD).
- The program attempts to establish a connection to the client TCP socket in an infinite loop. A five-second sleep interval is provided between each iteration to prevent CPU overload.

F. Dynamic Analysis of AsyncRAT

Process Monitor or Procmon is a feature-rich utility that allows efficient tracking and logging of all system activity, including file system, registry, process, and class activity. Based on Procmon's startup and execution times, the execution takes about four seconds as indicated in Fig. 5. The malware

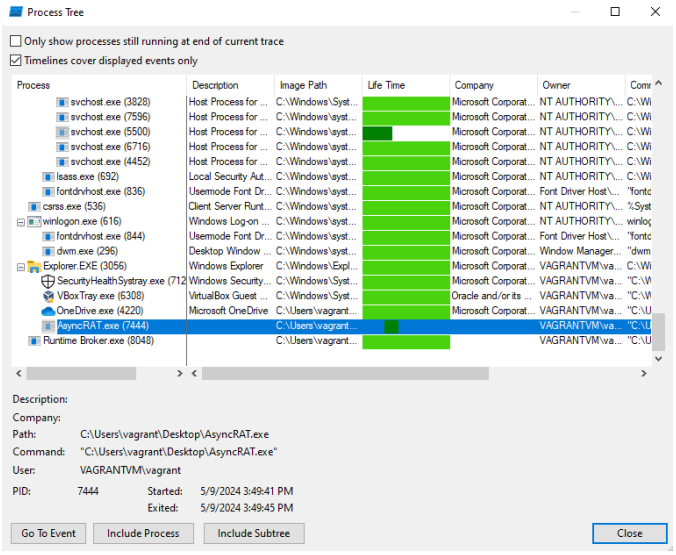

Fig. 5. Execution Time of the AsyncRAT Sample

also makes multiple attempts to connect to the IP address 217.195.197.70. This IP address seems to be linked with the attacker's C2 server(command and control server). Attempts to reconnect indicate that the virus is trying to establish a long-term connection with the controller. The malware creates these sockets to ensure secure network data transfer through TCP (Transmission Control Protocol) between the infected computer and the C2 server. This allows the attacker to send and receive data from the compromised computer. According to a VirusTotal search, the IP address turned out to be a malicious IP address from Turkey, as illustrated in Fig. 6.

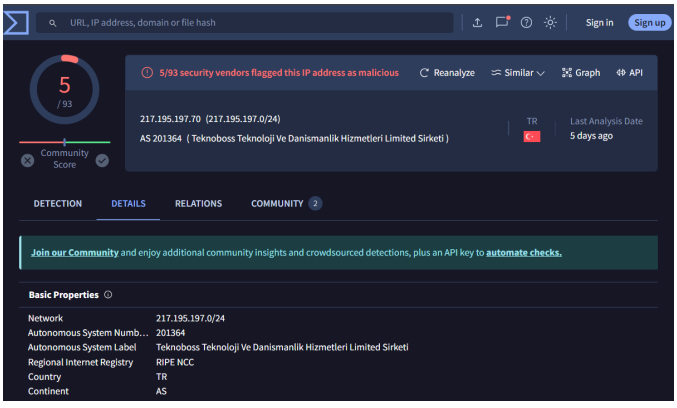

Fig. 6. A Malicious IP Located in Turkey

IV. Results and Analysis

A. Trickbot

The research unveiled numerous harmful actions associated with the TrickBot malware, a sophisticated and enduring threat. The static analysis of TrickBot necessitated various investigative tools and techniques to understand its characteristics and behaviors. Using VirusTotal, it was found that 29 out of 70 antivirus engines identified the TrickBot sample as malicious based on its MD5 hash, underscoring its harmful nature. Detect It Easy revealed significant characteristics of the sample, including its PE32 format, 5.51 MB size, and a timestamp of November 4, 2021 23:00:03. Further investigation evaluated the entropy of the TrickBot specimen, resulting in a substantial value of 6.39512, hinting at the use of encryption or packing methods for hiding. This high entropy suggests a complex design aimed at evasion. Moreover, extracting strings with Detect It Easy identified elements in the TrickBot binary that could reveal its operation, such as command and control server addresses, file paths, and registry keys. However, the presence of encrypted strings posed challenges to the analysis. Generally, these findings underscore the sophisticated obfuscation strategies employed by TrickBot to dodge detection and scrutiny.

B. AsyncRAT

The analysis of the AsyncRAT malware exposed some salient features of its behaviour and architecture. Its beginnings are connected to the APT-C-36 cyber organization, also known as Blind Eagle. This group is thought to have started in South America and has been active since April 2018. It was discovered that the malware had several features, such as remote desktop control and keylogging, which put victims at serious risk. Static analysis of AsyncRAT showed that the malicious software is a PE32 executable file made for 32-bit Windows OS, created with Visual Basic .NET, and connected with Microsoft Linker 8.0.

The entropy analysis revealed a moderate entropy value, suggesting that it lacks advanced encryption or obfuscation. A virus total search showed many AV vendors recognize the sample as a security threat. The ATT&CK Tactic and Technique analysis revealed that the malware employs multiple methods of data collection, defence evasion, discovery, execution, and persistence.

Through code analysis, it was discovered that AsyncRAT employs multiple methods to circumvent AV detection and avoid removal via AV/EDR. The live analysis by running the sample required around four seconds to complete. Throughout this period, the malicious software generates .exe and .bat files and tries to establish numerous connections to an IP address which is linked to a command and control (C2) server located in Turkey.

Analysis of network activity revealed that AsyncRAT creates TCP socket connections to communicate with the C2 server, enabling remote manipulation of the compromised system. This continuous attempt to establish a connection suggests that the malicious software is specifically created for the attacker to have ongoing access and control.

A thorough examination of TrickBot and AsyncRAT highlights the importance of strong cybersecurity measures and detailed threat intelligence. Comprehending the complex actions and avoidance methods utilized by these malware versions helps security experts create better detection and mitigation plans. Mapping the TTPs to the MITRE ATT&CK framework allows for customizing cybersecurity defenses to combat specific threats, ultimately strengthening overall security against modern malware.

V. Conclusion

Analyses of TrickBot and AsyncRAT reveal the intricacy and the threat these malevolent applications may pose. TrickBot employs a variety of tactics to gain access and maintain control, including spear-phishing campaigns for initial access, exploiting system vulnerabilities, and using a modular architecture for extensive functionality. TrickBot utilizes high entropy values to evade detection, and persistence mechanisms such as scheduled tasks to maintain long-term control over infected systems.

With the use of AsyncRAT, a powerful remote access Trojan, hackers may get significant remote access and control over target computers. Its two purposes are remote desktop control and keylogging. Many tactics are used for persistence, data collection, protected escape, and execution. Using a suspicious IP address to establish a connection to a C2 server in Turkey emphasises the potential risk even more.

These results highlight the necessity of developing strong defence strategies against such advanced malware. These shed light on important insights that contribute significantly to the continued development of cybersecurity solutions and mitigation of such threats, and serve as a base for future research and countermeasures.

A. Future Research Enhancements

Even though this paper has made a very profound analysis of TrickBot to AsyncRAT, the following are some future research directions:

- Behavioral Analysis through Machine Learning: This makes it easier to analyze malware behavioral patterns to anticipate or even identify new threats.
- Integration with Threat Intelligence Platforms: A tighter integration of analysis tools with threat intelligence platforms so that real-time updates on emerging threats occur for the improvement of detection and response.
- Research into new attack vectors: Research on the latest technologies like IoT, 5G, or AI, to see how the malware could exploit them and accordingly develop corresponding defensive strategies against the same.

With these research directions, better defenses can be put in place by the cybersecurity community against ever-evolving threats. In fact, substantial improvement can be made in the cybersecurity solutions to stop advanced malware threats.

REFERENCES

[1] Zhang, S., Wu, J., Zhang, M. and Yang, W., Dynamic Malware Analysis Based on API Sequence Semantic Fusion. Appl. Sci. 2023, 13, 6526. https://doi.org/10.3390/app13116526

[2] S. Al-Sofyani, A. Alelayani, F. Al-zahrani and R. Monshi, "A Survey of Malware Forensics Analysis Techniques and Tools," 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC), Jeddah, Saudi Arabia, 2023, pp. 1-6, doi: 10.1109/ICAISC56366.2023.10085474.

[3] Riya Nair, Kiranbhai R Dodiya and Parth Lakhalani, "A Static Approach for Malware Analysis: A Guide to Analysis Tools and Techniques.", 2023-12-20, https://www.ijraset.com/research-paper/a-static-approach-for-malware-analysis-a-guide-to-analysis-tools-and-techniques.

[4] Vishal Thakur, "Trickbot -a concise treatise.", blog, April 2019. Available: https://www.researchgate.net/publication/363737362_Trickbot_-a_concise_treatise

[5] Vishal Thakur, "TrickBot Execution Flow.", blog, February 2019, Available: https://www.researchgate.net/publication/363737342_TrickBot_Execution_Flow.

[6] Usman Sikander, "Unveiling the Intricacies of AsyncRAT: A deployment in Colombia by the Blind Eagle Cyber Group.", blog, Jan. 8, 2024. Available: https://medium.com/@merasor07/unveiling-the-intricacies-of-asyncrat-a-deployment-in-colombia-by-the-blind-eagle-cyber-group-83b48cc415a7

[7] Divya, T., Amritha, P.P. and Viswanathan, S., 2022. A model to detect domain names generated by DGA malware. Procedia Computer Science, 215, pp.403-412.

[8] Murali, R. and Velayutham, C.S., 2020, February. A conceptual direction on automatically evolving computer malware using genetic and evolutionary algorithms. In 2020 International Conference on Inventive Computation Technologies (ICICT) (pp. 226-229). IEEE.

[9] A. M. Bhatia, I. Kumar and N. Mohd, "Dynamic Analysis of a Malware Sample: Recognizing its behaviour using Forensic Application," 2023 4th IEEE Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2023, pp. 1-6, doi: 10.1109/GCAT59970.2023.10353478.

[10] Sinha, A.K. and Sai, S., 2023, July. Integrated Malware Analysis Sandbox for Static and Dynamic Analysis. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

[11] Ismael, M.F. and Thanoon, K.H., 2022, November. Investigation Malware Analysis Depend on Reverse Engineering. In 2022 International Conference on Data Science and Intelligent Computing (ICDSIC) (pp. 251-256). IEEE.

[12] Rohini, S., Ramesh, G. and Nair, A.R., 2024. MAGIC: Malware behaviour analysis and impact quantification through signature co-occurrence and regression. Computers & Security, 139, p.103735.

[13] Balasubramanian, K.M., Vasudevan, S.V., Thangavel, S.K., Kumar, G., Srinivasan, K., Tibrewal, A. and Vajipayajula, S., 2023, July. Obfuscated Malware detection using Machine Learning models. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-8). IEEE.

[14] Mithun, S., Abhinand, N., Pavithran, V. and Chandran, S., 2023, December. Stalkerware Detection Using Static and Dynamic Analysis. In 2023 IEEE 20th India Council International Conference (INDICON) (pp. 322-328). IEEE.

[15] Gregory Paul, T.G. and Gireesh Kumar, T., 2017. A framework for dynamic malware analysis based on behaviour artifacts. In Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications: FICTA 2016, Volume 1 (pp. 551-559). Springer Singapore.