MALWARE ANALYSIS AND INCIDENT RESPONSE USING ANY.RUN

SECURITY OPERATION CENTER (SOC)

NOT A BUSINESS CONFIDENTIAL

Prepared by: Gabriel D Ishengoma
Date: September 26th, 2024
Version: 1.1

ABSTRACT

This report, details an in-depth analysis of a malware sample known as ***"Remcos Remote Access Trojan"*** (RAT). The goal of this exercise was to examine the malware's behavior and its potential impact on a Windows 10 system. Using the ANY.RUN sandbox, a powerful tool that allows malware to run in a controlled environment, we were able to see how the Remcos RAT infects the system, changes settings, communicates with a Command and Control (C2) server, and attempts to hide its presence by embedding itself deeply into the system.

Remcos RAT is commonly distributed via phishing emails, where attackers trick users into downloading and opening infected files. Once activated, the malware establishes remote control over the target machine, which can result in dangerous actions like capturing what the user types *(keylogging)*, stealing sensitive information, or installing other malicious software. This report outlines the step-by-step process used to analyze the malware, from setting up the virtual machine *(VM)* to running the malware and observing its behavior. A comprehensive incident response plan was also created to address how to detect, contain, and remove the malware from the infected system.

The findings emphasize the importance of proactive measures in malware detection and system defense. The report concludes with recommendations to prevent such infections, improve security measures, and enhance response capabilities.

INTRODUCTION

In todays interconnected digital world, malware remains one of the most persistent threats, continuously evolving to bypass traditional security measures and infiltrate systems across the globe.
Malware is malicious software designed to damage, disrupt, or gain unauthorized access to computer systems, often with the goal of stealing data, spying on users, or enabling remote control over the infected machine. As cybersecurity professionals, it's crucial to understand how these threats operate and how they can infiltrate systems in order to defend against them effectively.

One of the most common forms of malware is the Remote Access Trojan (RAT), which allows attackers to take complete control over a victim's machine remotely. Once installed, a RAT can monitor user activity, steal sensitive information, deploy further malware, or even turn the infected device into part of a larger botnet. Malware like Remcos RAT can be particularly dangerous because it often goes unnoticed, running silently in the background while carrying out malicious tasks.

Delivery Methods of Malware
Malware can be delivered to a target machine through several common methods. These methods often exploit human behavior or vulnerabilities in software systems. Below are some of the most frequently used techniques for malware delivery:

Email Attachments One of the oldest and still most effective methods is through phishing emails, where the malware is disguised as a legitimate attachment. Users are tricked into opening documents, compressed files like *.zip* or *.rar*, or even executable files, which immediately infect their system upon opening. In this case, an archive file like winrar.exe could be part of such an attachment, where the user believes they are opening a harmless document or installer.

Malicious Links Attackers frequently use social engineering to lure users into clicking on malicious links embedded in emails, social media messages, or websites. Once clicked, these links either download malware automatically or direct the victim to a compromised website that exploits browser vulnerabilities to install malware in the background.

Infected Software or Cracked Applications Many users unknowingly download malware disguised as legitimate software. This is especially common with pirated or cracked versions of paid software. Attackers bundle malware with popular applications, hoping users will execute the infected package, thinking they're getting free software.

Drive-by Downloads Malicious websites or compromised legitimate sites can exploit browser vulnerabilities to deliver malware without any user interaction. These are known as drive-by downloads. Simply visiting an infected site could result in the stealthy download and execution of malware.

Malvertising (Malicious Advertisements) Cybercriminals place malicious advertisements on popular websites. Clicking on these ads, or even just visiting a page where the ad is hosted, can lead to a download or redirect to a malicious site that infects the system.

Removable Media (USB Drives) Malware can also be distributed through infected USB drives or other removable storage devices. Attackers leave infected USBs in public spaces, counting on unsuspecting users to plug them into their machines, thereby activating the malware.

Exploiting Vulnerabilities in Software Cybercriminals often take advantage of unpatched vulnerabilities in operating systems or common software applications. By crafting exploit kits that target these vulnerabilities, attackers can deliver malware to machines running outdated or unpatched software versions without any user interaction.

Once delivered, malware like Remcos RAT can embed itself deeply within the system, often by using system tools like PowerShell to download and execute further malicious commands. As illustrated in the behavior graph *(we will discuss more on page 11)*, the malware uses legitimate-looking processes such as winrar.exe to avoid detection, making it difficult for users to recognize the infection. It may then execute through PowerShell, spawn processes like conhost.exe and werfault.exe, and attempt to establish persistence to ensure it remains on the system even after reboots.

PREREQUISITES

Before starting, make sure you have these things ready

1. A Windows 10 virtual machine *(VM)* installed and ready to use.
2. Access to the *ANY.RUN* Sandbox.
3. A sample of the malware (in this case, Remcos RAT). Downloaded from Mali-ware Bazaar (*bazaar.abuse.ch*)
4. Basic knowledge of Remote Access Trojans (RATs), keylogging (capturing keyboard input), system registry settings, startup configurations, and how to analyze network traffic (the data sent and received over the internet).

IMPORTANT NOTE

This document is a brief analysis of malware behavior based on a real-world sample. It is intended to provide a simplified, high-level understanding of the malware execution flow and common delivery methods. Please note that this is not a formal Security Operations Center (SOC) report, and as such, it does not strictly adhere to the standard reporting structure and formalities typically required for professional SOC documentation. Instead, it is designed as an **informative guide** for educational purposes. The goal here is to simplify the complex malware behavior and provide practical insights to help both beginners and professionals understand how malware behaves.

DISCLAIMER

The malware samples available from Malware Bazaar are real and may contain actual malicious code. These samples are intended for research and educational purposes only. By accessing or downloading these samples, you acknowledge that you are aware of the risks involved and agree to take full responsibility for any consequences that may arise from their use.

Important Notes

- Do not execute these samples on your personal or production systems.
- Always use a controlled and isolated environment, such as a **virtual machine** designed for malware analysis.
- Ensure you have appropriate safeguards in place to prevent unintended harm to your systems or data.

By proceeding, you accept the inherent risks associated with handling malware.

CONTACT INFORMATIONS

Name	Title	Contact Information
Gabriel D Ishengoma	SOC & PenTester	Email: gabriel.ishengoma29@gmail.com
Abel Masanja	PenTester	Email: masanjaabel88@gmail.com

STEP BY STEP ANALYSIS

Step One: Set Up the Virtual Machine (VM)

Start by launching a Windows 10 virtual machine. A VM is used to ensure that the analysis happens in a completely isolated environment. This is critical when working with malware because it prevents the malware from escaping into your real operating system and network.

In our case, a freshly installed Windows 10 VM is preferable because it provides a clean slate with no prior modifications, making it easier to detect the changes that the malware makes to the system.

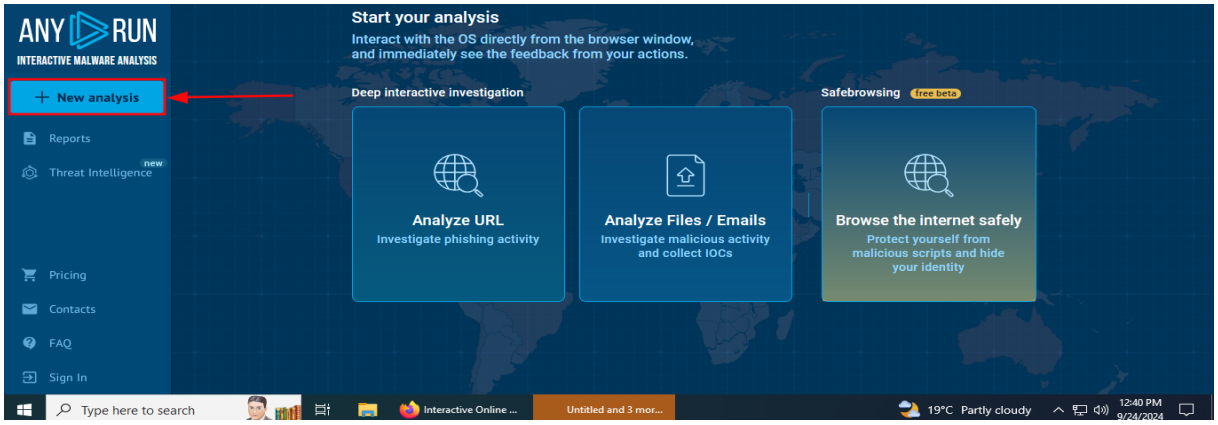
Step Two: Log in to ANY.RUN and Start a New Analysis

Once your VM is up and running, open a browser and go to ANY.RUN is a sandbox service that allows you to upload malware samples and observe how they behave in real time.

If you don't have an account create it because they offer free access lifetime *(but you can run every malware for 60sec per malware)* and then login.

After logging into ANY.RUN, click the *"+ New Analysis"* button to start.

You will be prompted to upload the malware file *(in this case, the Remcos RAT)* and select the operating system (Windows 10) in which you want to run the malware.



Important: Selecting the right OS is crucial because malware often behaves differently on different operating systems.

Step Three: Static Analysis

Before running the malware, we performed a **static analysis**. Static analysis is like "*examining a box without opening it*". We're not running the malware yet, instead, we are gathering basic information about the file itself.

File info:	PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows
MD5:	FF6F202ACE40743852A03F34B7B41707
SHA1:	31E96D82538ECD77F5A190BBC070065CB64BD12B
SHA256:	66C50343775C162862AC27A735C66927A9B3FDA4A05CD0EAA21FECBCA3F6C490

Here are the key pieces of information we collected from the malware sample

- **File Type** The malware is a *PE32 executable*. This means it's a 32-bit Windows executable file. Most Windows programs are in this format.
- **MD5 Hash** The MD5 hash is a unique fingerprint of the file. No two files with different content will have the same MD5 hash so this is a helpful way to identify the malware.
- **SHA1 Hash** Similar to the MD5 hash the SHA1 hash is another way of identifying the file.
- **SHA256 Hash** The SHA256 hash provides an even stronger and longer fingerprint that making it nearly impossible for two different files to have the same hash.

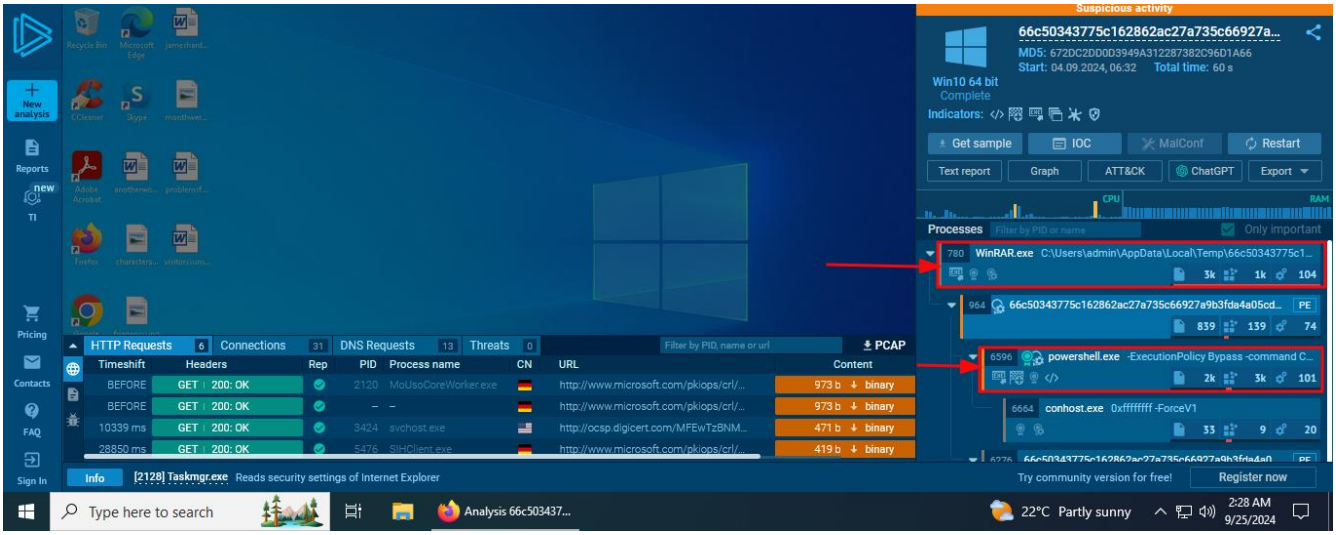
We can also use a tool called **IDA Pro, Ghidra or Binary Ninja** for reverse engineering the malware. Reverse engineering involves converting the malware binary code back into readable assembly language so we can understand what the program is designed to do. This process revealed that the malware was trying to install itself in the system registry and start automatically when the computer is turned on. *(For this case we will focus on Dynamic analysis using ANY.RUN)*

Step Four: Dynamic Analysis

After completing the static analysis, we moved on to **dynamic analysis**. This is where we actually run the malware in the ANY.RUN sandbox and observe how it interacts with the system.

Upon execution, we immediately noticed several **suspicious processes**

- **WinRAR.exe** The malware used this process to extract files and make changes to the system. This is another indication of potentially harmful activity.
- **powershell.exe** This process is commonly used by attackers to execute scripts and commands without raising alarms. It is concerning to see it being used here because it suggests the malware is using PowerShell to run malicious scripts.



File Modifications

The malware also made several changes to system files. These modifications were carried out by PowerShell and WinRAR. Such changes often point to attempts at **persistence**, which means the malware is trying to make sure it stays on the system even after the computer is restarted.

Network Activity and Connections

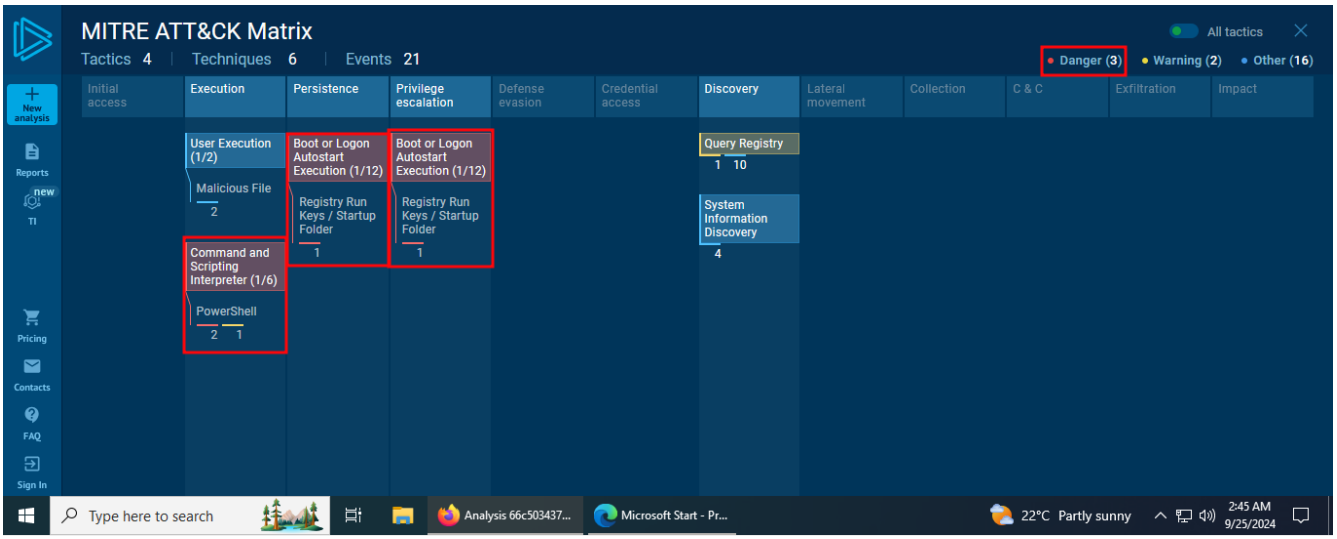
Among of the most alarming findings was the network activity, malicious connection to external IP addresses that generated by the malware. It initiated several **HTTP requests and Connections** to a remote server. These requests are part of the communication between the infected machine and a Command and Control (C2) server. A C2 server is controlled by the attacker and is used to send commands to the malware on the victim computer as well as to receive stolen data.

This communication is a clear sign that the infected machine could be controlled remotely and making it vulnerable to further attacks like data theft or the installation of additional malware.

MITRE ATT&CK Framework

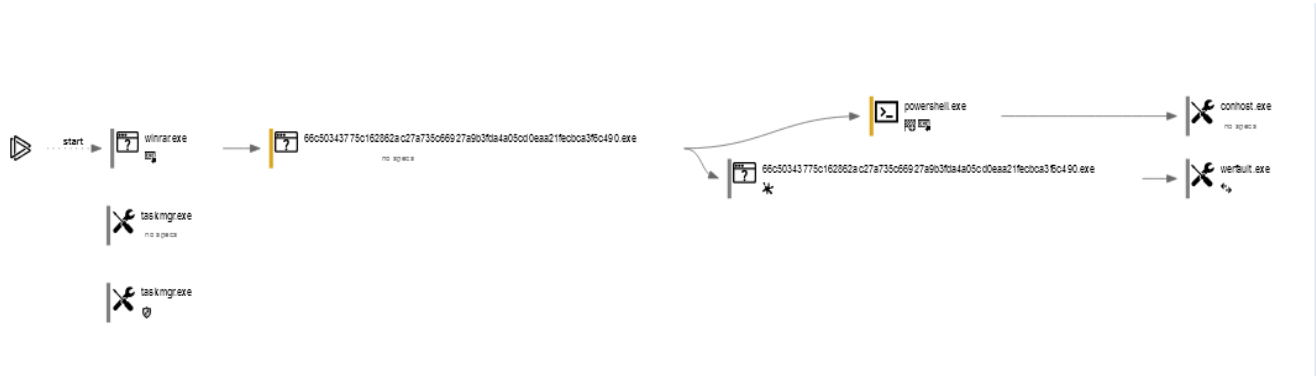
Using the **MITRE ATT&CK** framework, we were able to categorize the tactics used by the malware. This framework helps us understand how malware behaves and what it aims to achieve. In this case, the Remcos RAT employed the following tactics

1. **Execution** The malware executes its code to gain control of the system.
2. **Persistence** It modifies the system to ensure that it remains active even after the computer restarts.
3. **Privilege Escalation** The malware attempts to gain higher level permissions to perform actions that a normal user would not be allowed to do.



Behavior of the malware

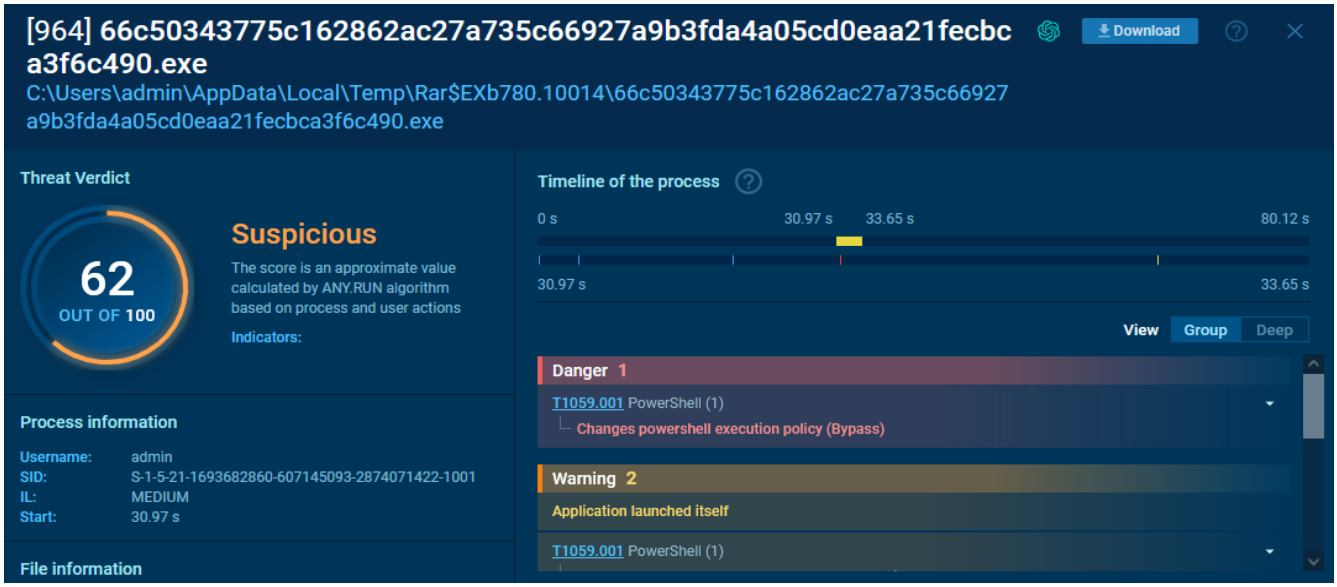
The following diagram (process graph) shows the flow of malware execution and the processes it triggered during our analysis:

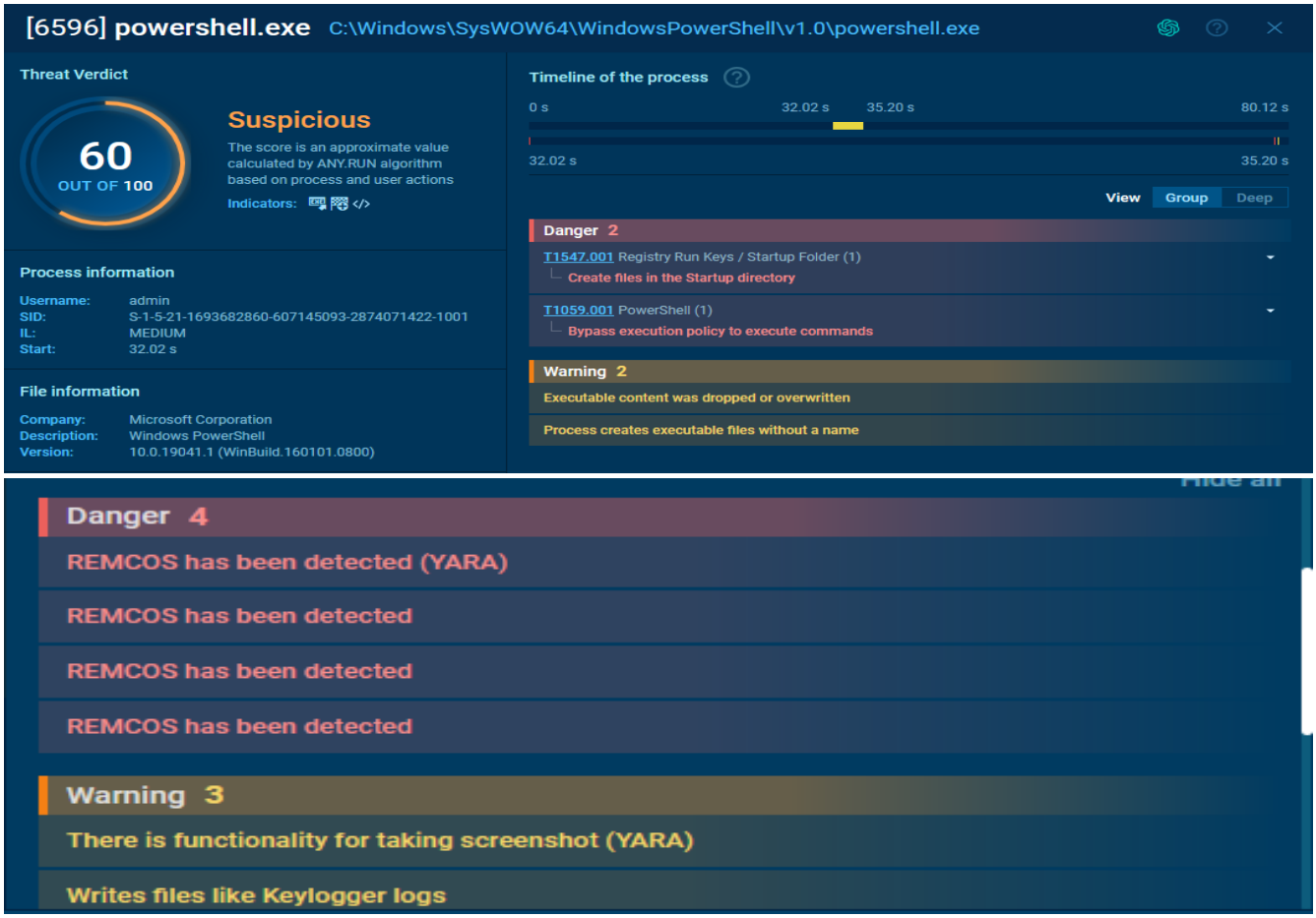


i. The process begins with the execution of winrar.exe, which is a legitimate application used for compressing or extracting files. In this case, the malware might be hiding inside an archive file, and when extracted or run, the malware process is triggered through **winrar.exe.**

ii. The suspicious file invokes **powershell.exe**, a command-line shell commonly misused by malware to execute harmful scripts or commands. This is a significant indicator of malicious activity because PowerShell can be used to download payloads, execute hidden commands, or communicate with remote servers without user knowledge.

iii. The creation of **conhost.exe** and **werfault.exe** is suspicious, indicating the malware may be trying to hide its activity under legitimate system processes.





Results and Discussion

The results of our analysis paint a clear picture of how dangerous the Remcos RAT can be. Some of the key takeaways are

- **File Modifications** The malware modified several critical files on the system. It used PowerShell to execute malicious commands and WinRAR to make changes to system files. These modifications are part of the malware's attempt to stay hidden and maintain persistence.
- **HTTP Requests and C2 Communication** The malware-initiated HTTP requests to a remote server. This is one of the clearest signs of a Remote Access Trojan, as it confirms that the infected machine is being controlled remotely. The C2 server can send commands to the malware, which could include stealing data, capturing screenshots, or logging keystrokes.
- **Persistence Mechanisms** The malware changed the system registry so that it would start automatically when the computer boots up. This ensures that even if the system is restarted, the malware will still be active and ready to receive commands from the C2 server.

Conclusion and Recommendations

This analysis demonstrates the highly invasive and dangerous nature of Remcos RAT. Its ability to modify system files, maintain persistence, and communicate with a remote C2 server makes it a serious threat to any system it infects. The key lessons learned from this analysis are the importance of detecting such threats early and responding quickly.

Here are our recommendations

1. **Email Security** Since this malware can be delivered through phishing emails, it is important to improve email filtering to block suspicious or harmful attachments before they reach users.
2. **Monitor Registry Changes** Regularly monitor the system registry for unusual changes, as this is one of the key areas malware targets for persistence.
3. **Network Traffic Analysis** Use network monitoring tools to track unusual outgoing connections. If you see HTTP requests going to unknown or suspicious servers, it could indicate a malware infection.
4. **Update Security Software** Ensure that antivirus and other security tools are regularly updated to recognize and block new malware variants like Remcos RAT.