Escuela Colombiana de Ingeniería Julio Garavito

Demo Company
IT Security and Privacy

Static Malware Analysis

Students:

Laura Sofia Gil Chaves

Camilo Castaño Quintanilla

Teacher:

Ing. Daniel Vela

Business Confidential

October 09th, 2024

Project 01-11

Version 1.0

Table of Contests

Assessment Overview

From Thursday, October 3 to Monday, October 7, static malware analysis is a fundamental technique for identifying vulnerabilities without executing the suspicious code, allowing a detailed review of its structure and behavior. In this case, the focus is on Silly Putty, a program that undergoes an exhaustive analysis process with the aim of discovering possible security holes.

1.Planning: Identify key actions to discover vulnerabilities in malware.

2.Discovery: The files are examined, and the code is reviewed in order to detect any behavior that we can explain.

3.Attacking: In this phase, the malware was analyzed in detail and its components were inspected.

4.Reporting: Documentation of vulnerability analysis and possible mitigation options.

Plan → Discovery → Attack →Report

Assessment Components

External Penetration Test

A network was created in VirtualBox with a different IP range than the university network, assigning IP ranges and configuring the DHCP server. The new network is then assigned to the virtual machine using a Host-Only Adapter, disabling the other adapters. Additionally, static analysis methods are combined using tools such as PEStudio, which performs a comprehensive analysis of the malware and correlates it with indicators in Virus Total. Finally, malicious capabilities are identified in binary files.

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Vulnerabilities that allow full exploitation of the system, resulting in the possibility of compromising all aspects of the system, with severe impact on security, data integrity and availability of critical services.
High	7.0-8.9	Vulnerabilities that facilitate unauthorized access to critical areas of the system, allowing the modification or theft of sensitive information, with a significant risk of impact on privacy and the general operation of the system.

Scope

Assessment	Details
External Penetration Test	GitHub - HuskyHacks/PMAT-labs: Labs for Practical Malware Analysis & Triage

Scope Exclusions

There will be no scope exclusions regarding static malware analysis, as all testing will be conducted in a secure and controlled laboratory environment. The primary focus will be on static malware analysis.

Client Allowances

The permissions required to perform the tests will be provided by the Husky Hacks page, which will allow full access to all necessary files and resources.

Executive Summary

The hash value of a file is a unique, compact representation of its contents, generated using a hash function such as SHA-256. This value, usually expressed in hexadecimal format, allows you to verify the integrity of the file. In this laboratory, the sha-256 command was implemented to study the PuTTY malware. Once the hash value is obtained, the total virus

program is implemented to detect attacks caused by the file. Finally, the study system is used to determine the environment of the file and what communication it causes with the machine.

Attack Summary

The following table describes how we gained internal network access, step by step:

Step	Action	Recommendation
1	Get the putty´s challenge (malware) from the GitHub repository HuskyHacks/PMAT-labs/tree/main/labs/1-1.BasicStaticAnalysis	Run the Putty challenge in an isolated virtual machine (VM) with no network access to prevent malware spread.
2	Call the sha256sum command on the malware executable.	Use .malz extension to prevent malware spread. In addition, avoid executing the file, and securely store the hash for future reference.
3	Open the program *Virus Total* to submit information about a virus.	Avoid submitting the actual malware executable; instead, submit the hash to prevent accidental execution and potential risks.
4	Extract and analyze the strings from the malware executable using PEStudio.	Always ensure your PEStudio is updated to the latest version before analysis. This helps improve detection capabilities and ensures you have the latest features for better insight into the malware.

Security Weaknesses

The following weaknesses that a system may have if it contains malware are described.

Weak Passwords

Simple passwords can be easily guessed by attackers. Implement strong password policies that require combinations of letters, numbers and special characters, along with two-factor authentication.

Out-of-Date Software

Vulnerabilities in out-of-date software are a common target for malware. Maintain an automated update management system that ensures all programs, and the operating system are kept up to date with the latest security patches.

Lack of Antivirus or Antimalware

The absence of protection software allows malware to be installed without detection. Install and maintain reliable antivirus and antimalware software configured to perform regular scans and automatic updates.

Unsecured Networks

Connecting to unprotected public Wi-Fi networks can facilitate access to sensitive data. Use a virtual private network to encrypt the connection and protect transmitted information, especially on untrusted networks.

Disabling Security

Disabling firewalls or access controls increases vulnerability. Ensure that security settings are applied and continuously monitored and avoid disabling security functions unless necessary and with adequate control.

External Penetration Test Findings

External Penetration Test Findings

Description:	Static analysis methods were combined using tools such as PEStudio, which performs a comprehensive analysis of Silly Putty malware and correlates it with indicators in Virus Total. Finally, malicious capabilities were identified in the binary files.
Impact:	Critical
System:	Windows
References:	GitHub - HuskyHacks/PMAT-labs: Labs for Practical Malware Analysis & Triage

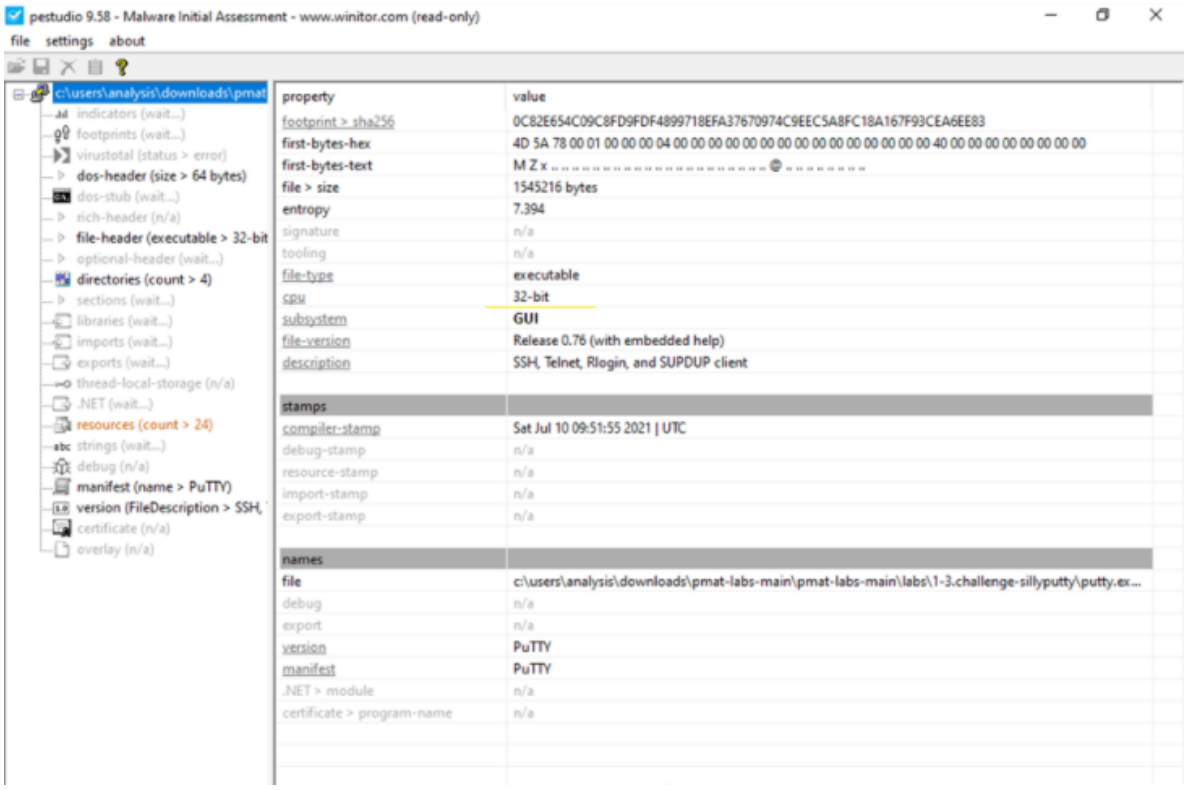
Exploit Proof of Concept

1. **What is the SHA256 hash of the sample?**

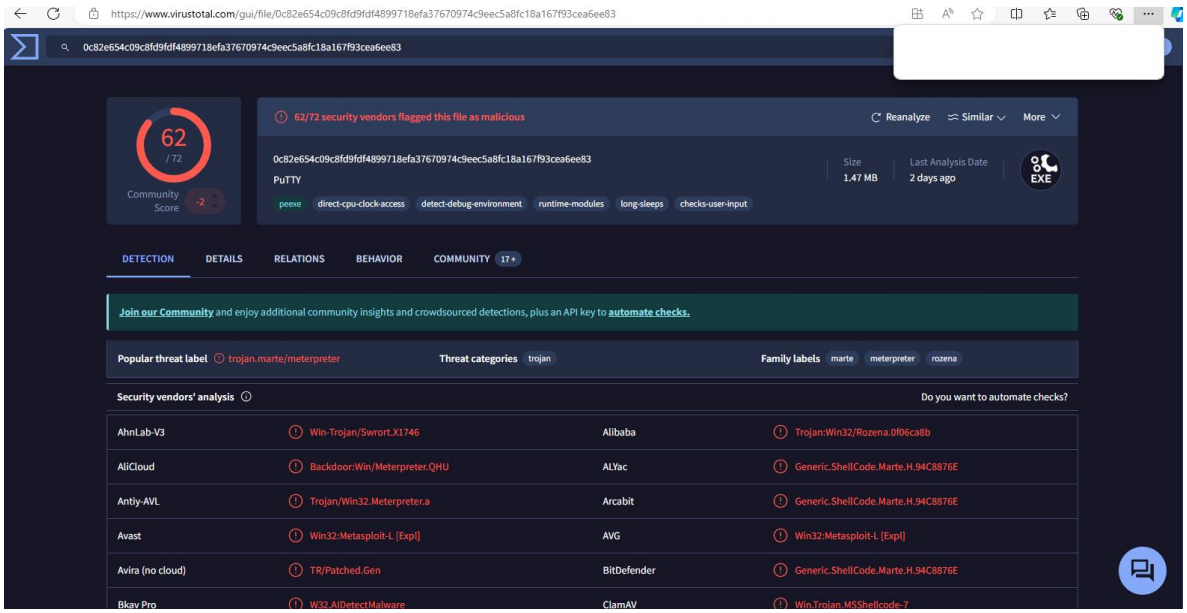
   The SHA256 hash is
   `0c82e654c09c8fd9fdf4899718efa37670974c9eec5a8fc18a167f93cea6ee83`

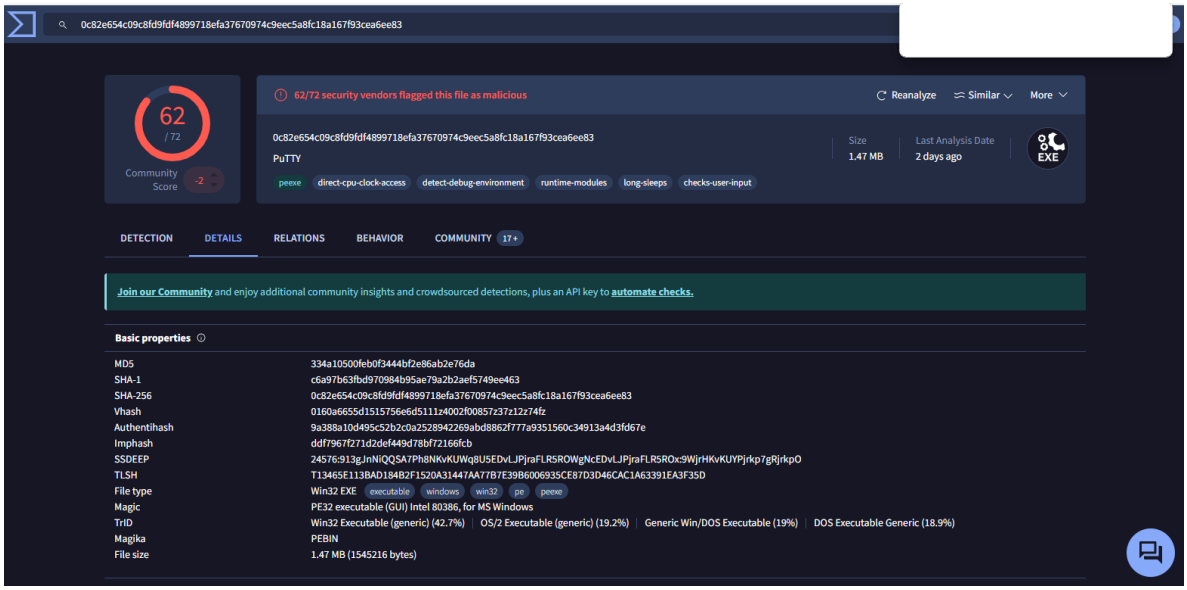
2. What architecture is this binary?

Analyzing the binary file with PEStudio identifies that it is a 32-bit binary. This means that it is designed to run in a 32-bit environment, which is relevant for its compatibility with operating systems and limits its access to a maximum of 4 GB of RAM.



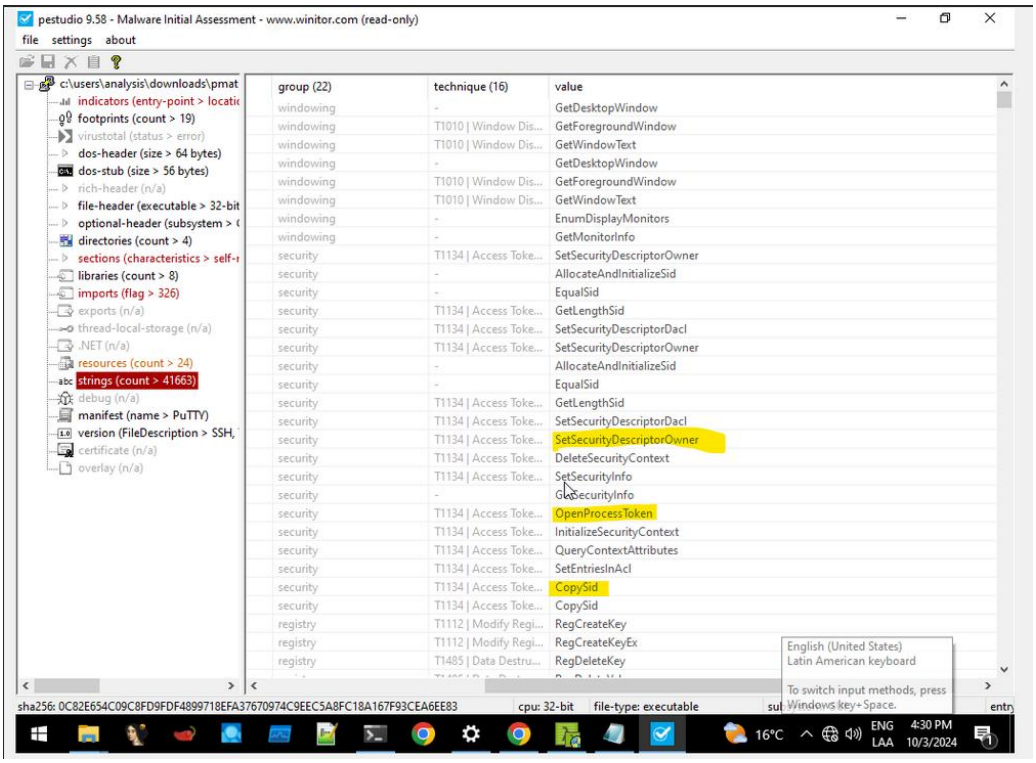
3. Are there any results from submitting the SHA256 hash to Virus Total?

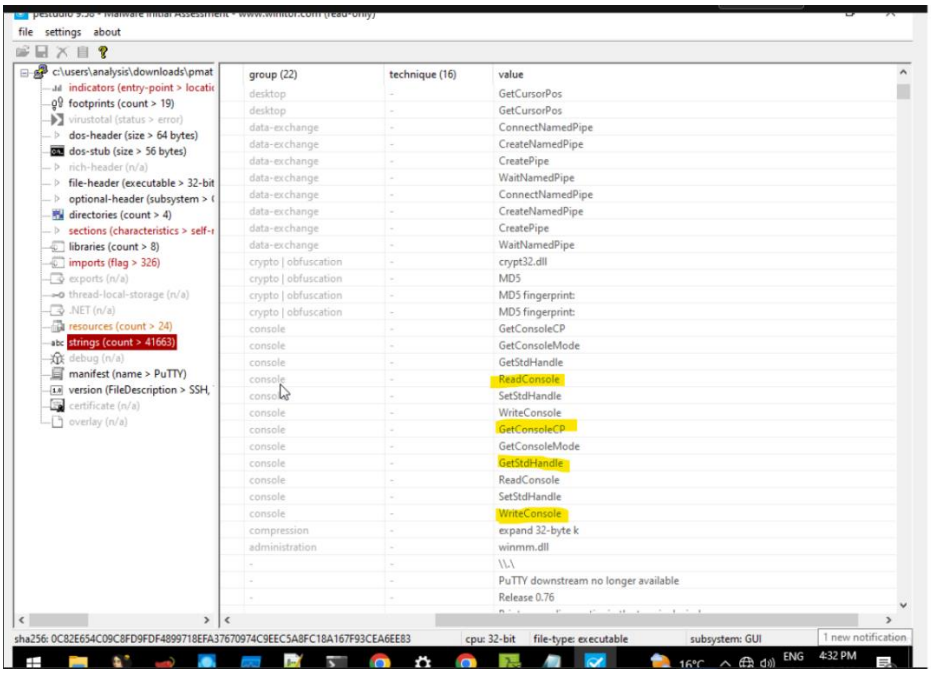




What we can observe with total virus is that the putty executable is 86% a virus, because almost all its properties are a Trojan, which is a type of malicious software that disguises itself as a legitimate program to deceive users. users and have them install it. Unlike viruses or worms, a Trojan does not replicate itself. Instead, it allows attackers to access and control the victim's system, steal information, install other malware, or carry out malicious activities without the user's knowledge.

4. **Describe the results of pulling the strings from this binary. Record and describe any strings that are potentially interesting. Can any interesting information be extracted from the strings?**





The functions GetConsoleCP, GetStdHandle, WriteConsole, and ReadConsole are related to console input and output operations in a Windows environment.

- **GetConsoleCP**: This function retrieves the current input code page for the console, which determines how characters are interpreted from input.

- **GetStdHandle**: This function obtains a handle to a standard device, such as the console's input or output stream, allowing the program to read from or write to these devices.
- **WriteConsole**: This function outputs data to the console, enabling the program to display information or messages to the user.
- **ReadConsole**: This function reads input from the console, allowing the program to capture user input directly.

Together, these functions suggest that the program interacts with the system's console, facilitating communication with the user through text-based input and output. This interaction is crucial for applications that require user input or need to display information in a command-line interface. In the other hand, the binary includes a manifest that references PuTTY, a widely used software for remote connections and SSH. This suggests that the malware may be associated with remote access or could be attempting to exploit legitimate tools like PuTTY to establish unauthorized connections.

Finally, several functions related to security management, such as SetSecurityDescriptorOwner, OpenProcessToken, and CopySid, are utilized to handle security identifiers (SIDs) and system privileges.

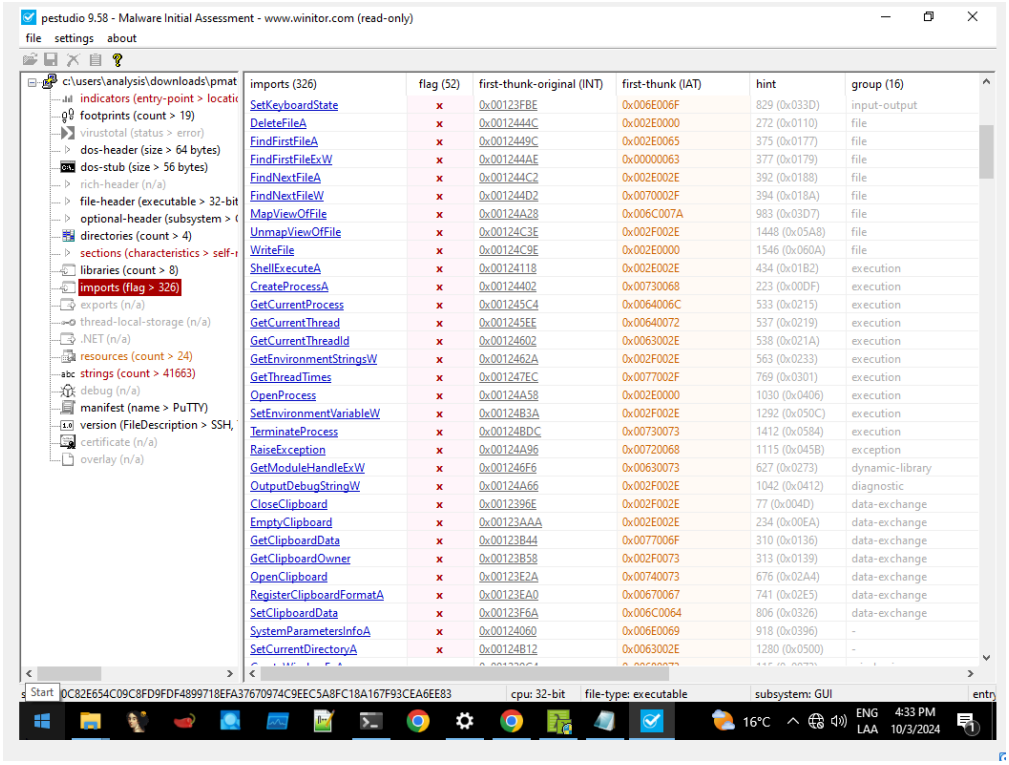
- **SetSecurityDescriptorOwner:** is a Windows API function used to set the owner of a security descriptor.
- **OpenProcessToken:** is a Windows API function that retrieves a handle to the access token associated with a specified process. Access tokens contain information about the security context of a process, including the user account that is running the process, the privileges it has, and its group memberships.
- **CopySid:** is a Windows API function used to create a copy of a Security Identifier (SID) structure. A SID uniquely identifies users or groups within Windows security.

This indicates that the malware may be attempting to manipulate security tokens or alter permissions in order to elevate its privileges within the system. By leveraging these functions, the malware could gain unauthorized access or control over protected resources.

5. **Describe the results of inspecting the IAT for this binary. Are there any imports worth noting?**

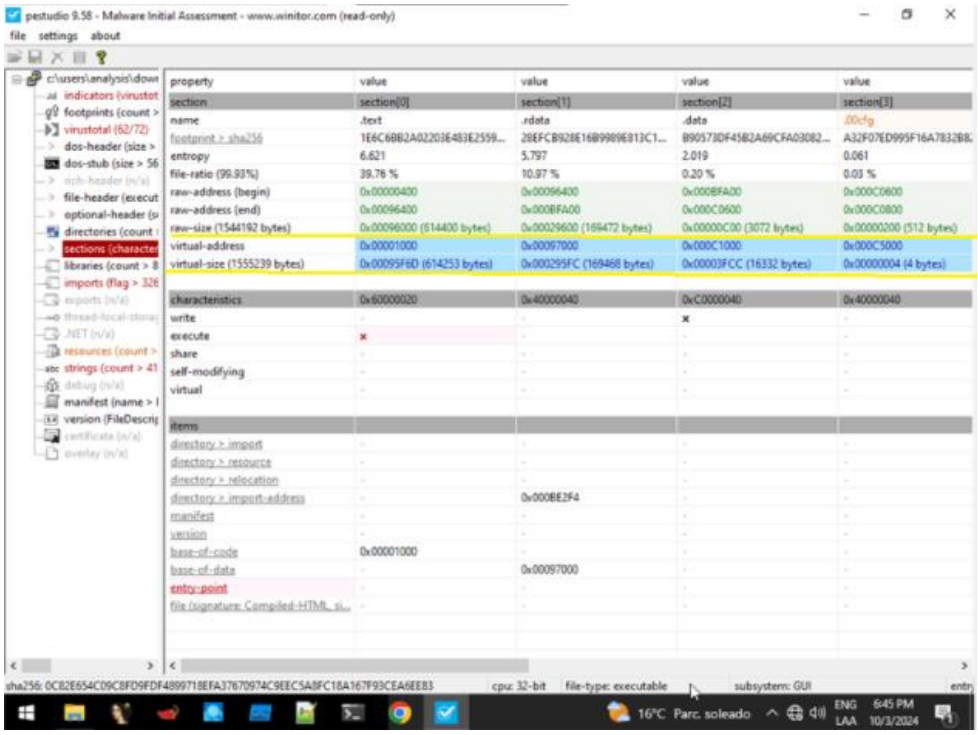
   When inspecting the Import Address Table (IAT) in PEStudio, imports related to the Windows Registry are observed, suggesting that the Silly Putty malware may have the ability to manipulate system settings. However, it is important to note that these functions can also be legitimately used by the program. Although the IAT presents

several imports that require further analysis, the lack of sufficient information prevents definitive conclusions about its malicious behavior.



6. Is it likely that this binary is packed?

It is unlikely that this binary is packed, since the values of the raw data size and virtual size of the headers are quite similar, supporting the conclusion that no packing has been applied to this file.



Recommendation:

Who:	The system security team and developers of the programs.
Vector:	Remote.
Action:	*Item 1*: the user should ensure that their operating system, applications and security software are always up to date with the latest patches and updates. This will help close vulnerabilities that can be exploited by malware. *Item 2*: the user should install and maintain a reliable anti-virus or anti-malware program. Performing regular scans and enabling real-time protection will allow threats to be detected and blocked before they can cause damage. *Item 3*: users should avoid opening emails from unknown senders and do not download suspicious attachments or links, as many malware infections are initiated through social engineering techniques. *Item 4*: the user should enable and configure a firewall on their system to monitor incoming and outgoing traffic. This can help block unauthorized access and detect unusual activity on the network.