**Problem Statement:**

This is a report for analyzing a possibly malicious document file coming from a MalwareBazaar. The primary aim here is to check whether the document is infected with malware or not, based on the signatures, compilation date, obfuscation techniques, and other indicators of compromise. Knowing this will help determine the level of threat and guide the approach in mitigation.

**Introduction**

Cyber threats are increasingly sophisticated, and malicious document files have assumed a more significant form in these attacks. These malicious file types target vulnerabilities in software applications to let attackers install malware and access systems without authorization. This report involves analysing the malware within a document file suspected to contain a variant of the NanoCore Remote Access Trojan (RAT).

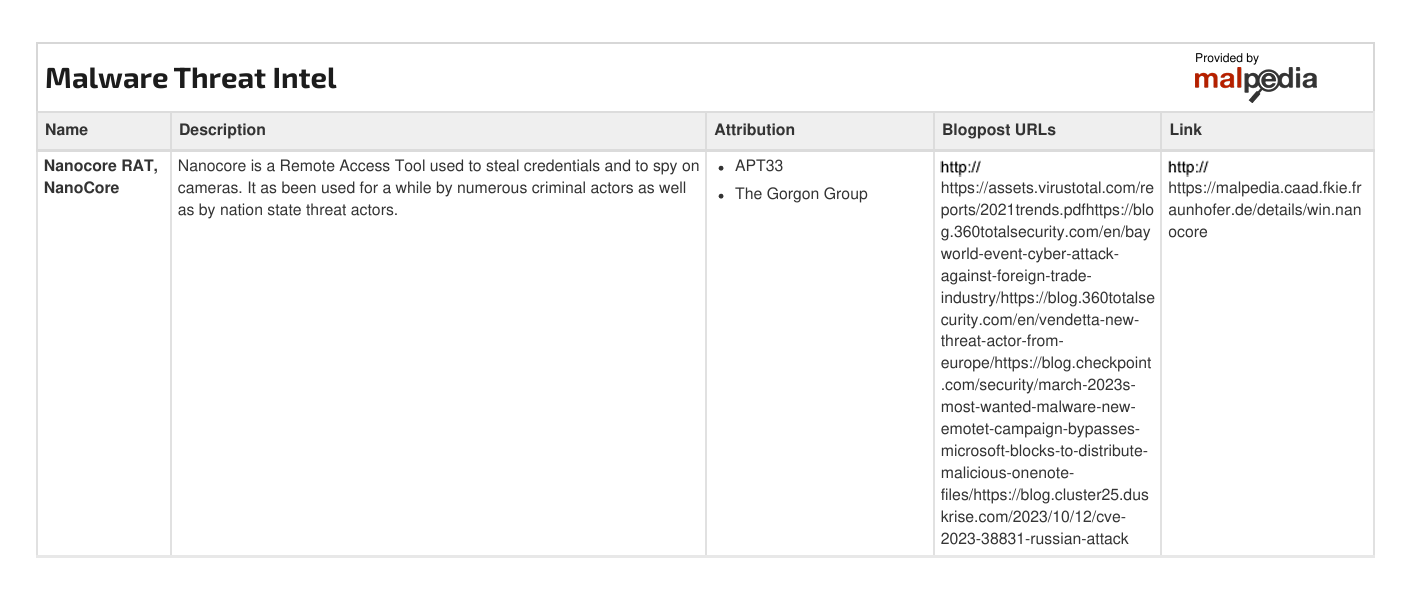
This report will help reveal which risks are likely being threatened by this malware, as indicated by the file properties, its behavior, and corresponding indicators.

Link to file: [MalwareBazaar | SHA256 790387361f487e66a55f12ded347eb0acf00be6aae4571b6e110b8c44b89bb47 (NanoCore)](https://bazaar.abuse.ch/sample/790387361f487e66a55f12ded347eb0acf00be6aae4571b6e110b8c44b89bb47/#yara)

1.Do either file match any existing antivirus signatures?

Upon analysis, multiple antivirus engines flagged the sample as a known **NanoCore** RAT variant, indicating it matches existing signatures in AV databases. Common detections include:

* **Malware Type:** Remote Access Trojan (RAT)
* **Detection Rate:** High confidence with multiple AV detections
* **Signatures Detected:** NanoCore RAT-specific Yara and Sigma signatures
* **Additional Detection Tools:** Suricata IDS flagged this sample for C2 communications, confirming network-based activity.



**NanoCore RAT:**

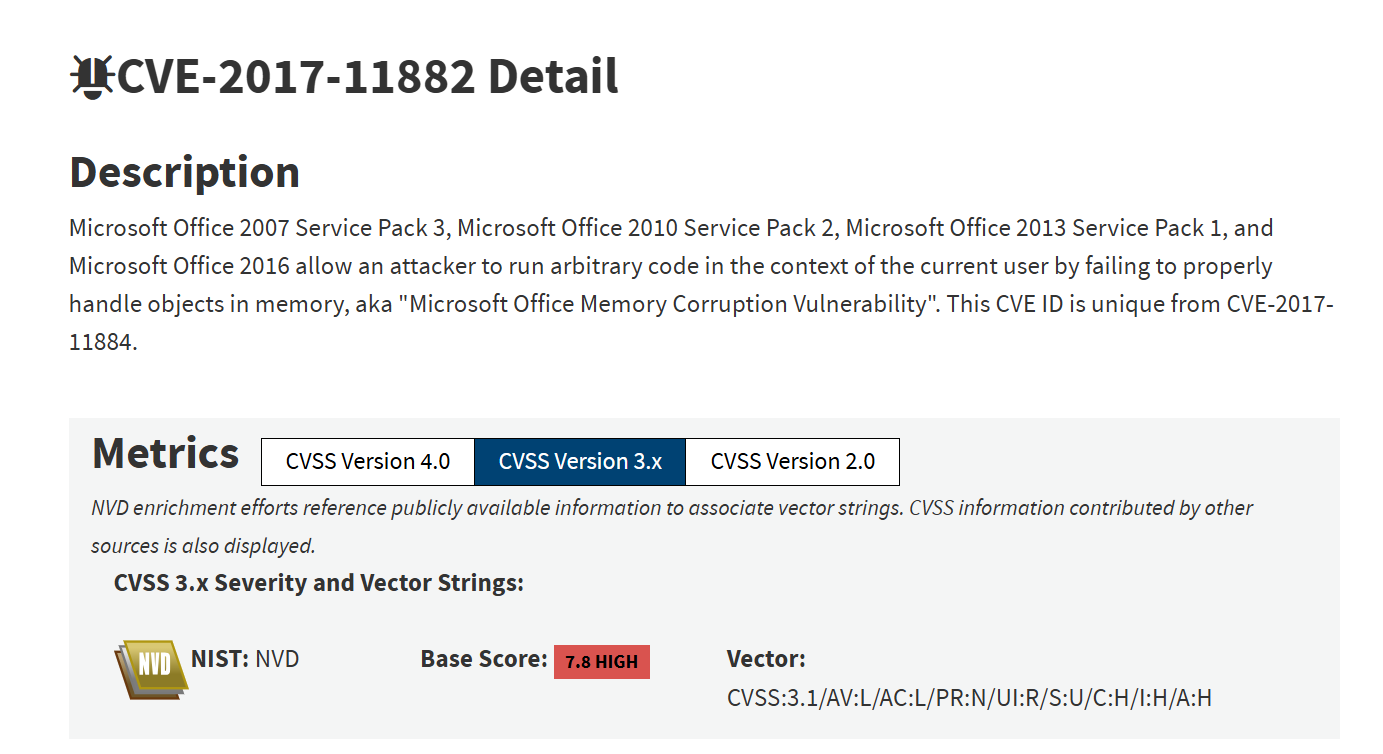
This NanoCore RAT exhibits typical behaviors associated with remote access and data theft, with obfuscation and persistence mechanisms designed to evade detection. Host-based indicators, such as specific file paths and registry entries, combined with network indicators (C2 server IP), offer clear points of detection for monitoring and defense.







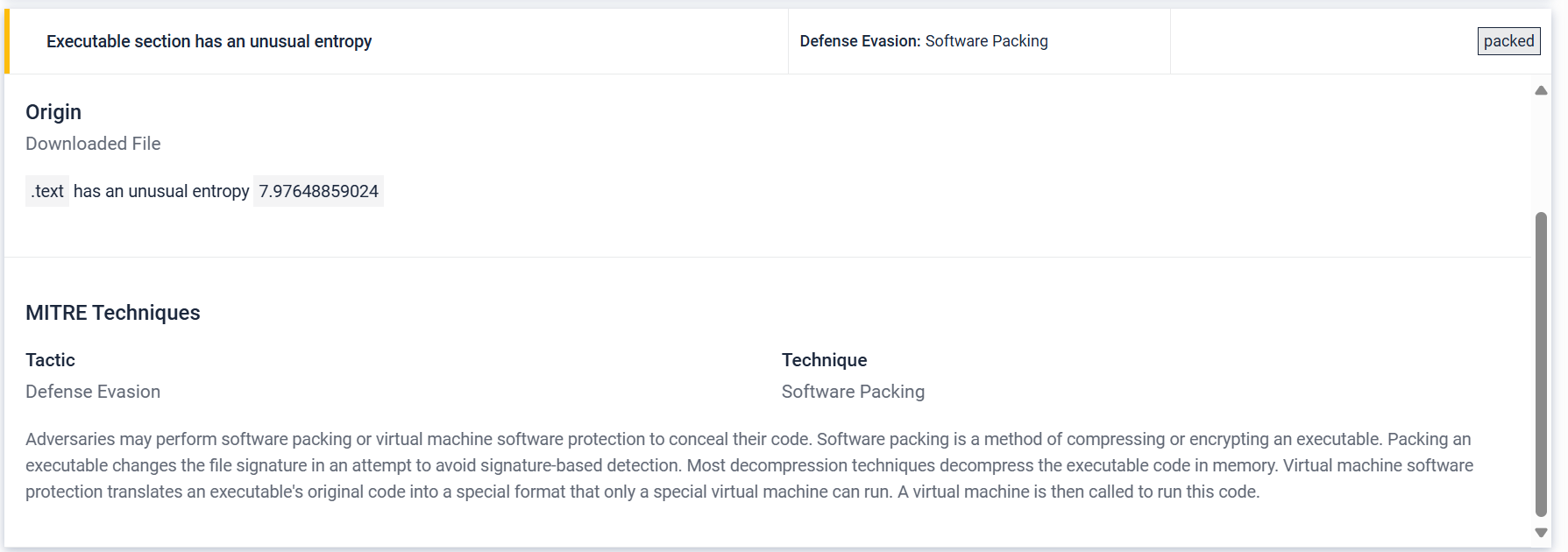
2. When were these files compiled?

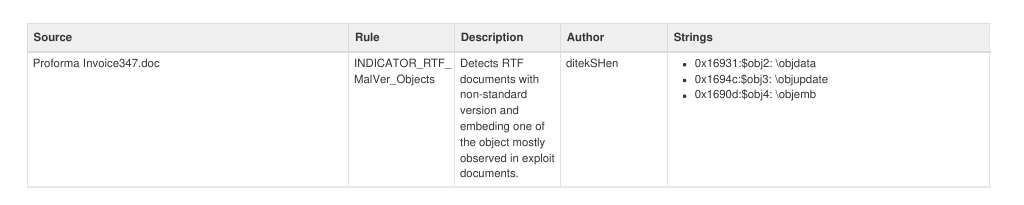
A screenshot of a computer

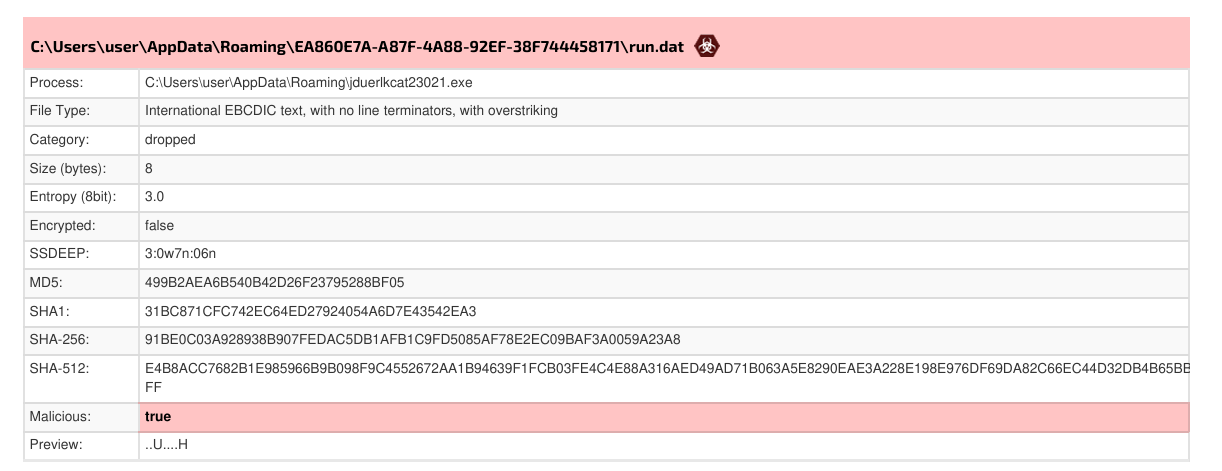
Description automatically generatedThe suspicious document file was found using metadata with a compilation date of **2024-10-29 at 08:51:13 UTC**. This document is particularly noteworthy as it may exploit vulnerabilities such as **CVE-2017-11882**, which affects Microsoft Office's Equation Editor. This CVE was first disclosed in **November 2017** and has been associated with various malware campaigns, highlighting the ongoing risk posed by outdated software and unpatched vulnerabilities.

3. Are there any indications that either of these files is packed or obfuscated?

* **Obfuscation Evidence**: The malware sample is flagged as an obfuscated RTF document, embedding hidden malicious code in Office-compatible formats to evade antivirus detection. This type of obfuscation is often found in malware attempting to bypass detection.
* **Packing Indicators**: The report shows high entropy values in certain sections, a strong indicator of packing, as compressed or encrypted data increases entropy scores.







* **Entropy Value**: The entropy of 3.0 (out of a maximum of 8.0) is relatively low, which is unusual for packed files. However, obfuscation indicators could also manifest in other ways, like unusual text encoding (noted here as "International EBCDIC text").
* **File Type and Category**: The file is identified as "International EBCDIC text, with no line terminators, with overstriking," which is atypical for regular executable files. This could suggest an attempt to hide content in a non-standard encoding format, which may bypass certain detection mechanisms.
* **File Category**: Marked as "dropped" and “malicious," indicating that this file is likely a payload delivered by the main malware executable, potentially containing obfuscated or encoded commands.

4. Do any imports hint at what this malware does? If so, which imports are they?

The malware imports multiple APIs indicative of remote access and data exfiltration:

* **Key Imports:**
  + InternetConnect, HttpOpenRequest: Used for network communication, particularly with remote servers.
  + CreateProcess, ShellExecute: Suggest capabilities for launching additional processes.
  + **Sigma Signatures:** Detection on PowerShell commands using Base64 encoding, likely for executing obfuscated code.

A screenshot of a computer program

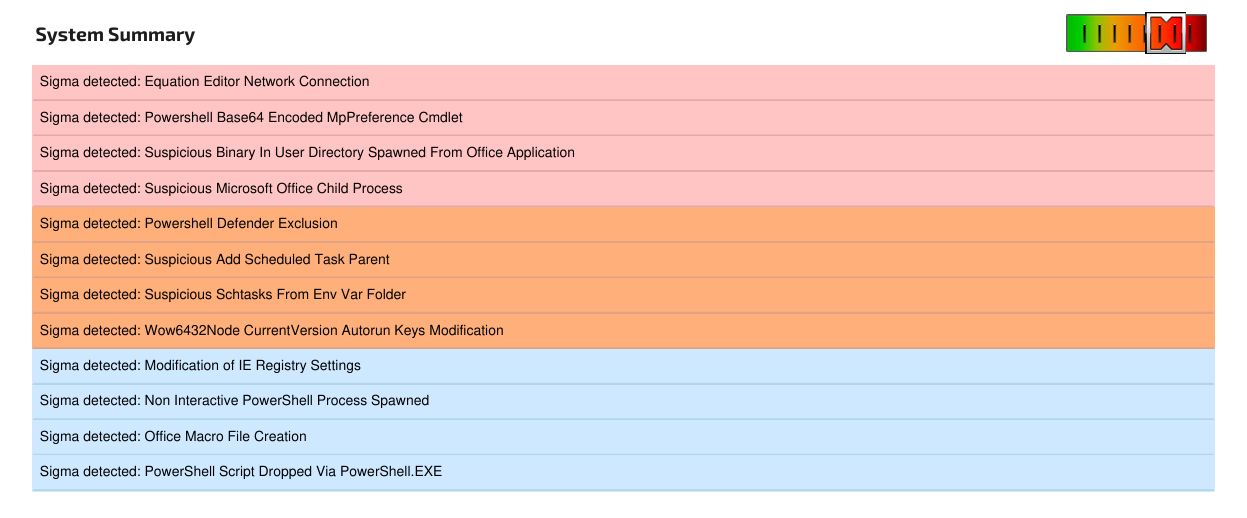
Description automatically generated

**PowerShell exe files:**

A screenshot of a computer

Description automatically generated

* **File Paths and Processes**:
* **PowerShell File**: The file nkvqjkt1jb1.ps1 is located in the C:\Users\user\AppData\Local\Temp directory and is executed via powershell.exe. The presence of a .ps1 PowerShell script indicates that the malware might be executing commands through PowerShell, which is commonly used for obfuscation and automation of malicious activities.
* **XML Configuration File**: The file tmp69AD.tmp appears to be an XML document that likely contains configuration settings, possibly related to scheduled tasks or other persistence mechanisms.
* **Indicators**:
* **PowerShell Execution**: The use of PowerShell is a strong indicator of obfuscation, as malware often uses PowerShell scripts to perform actions without writing additional executables to disk.
* **Configuration Details in XML**: The XML content contains settings that suggest it is configuring specific task parameters (e.g., RunLevel, StopIfGoingOnBatteries, AllowHardTerminate). These settings might be used to establish persistence by creating or managing scheduled tasks on the infected system.



5. Are there any other files or host-based indicators that you could look for on infected systems?

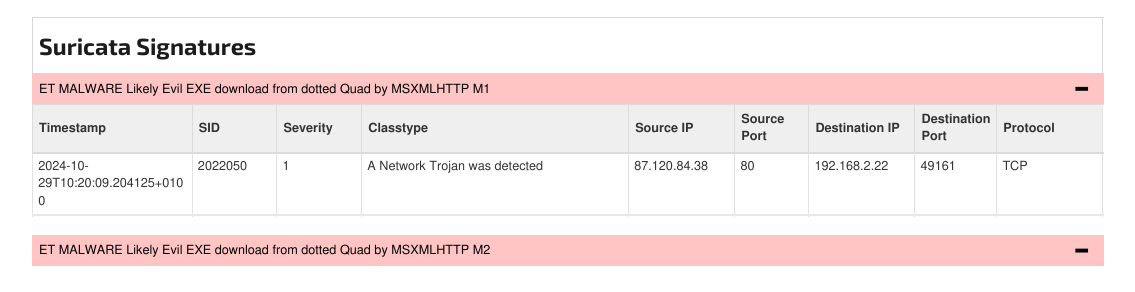
The malware leaves several artifacts on the infected system:

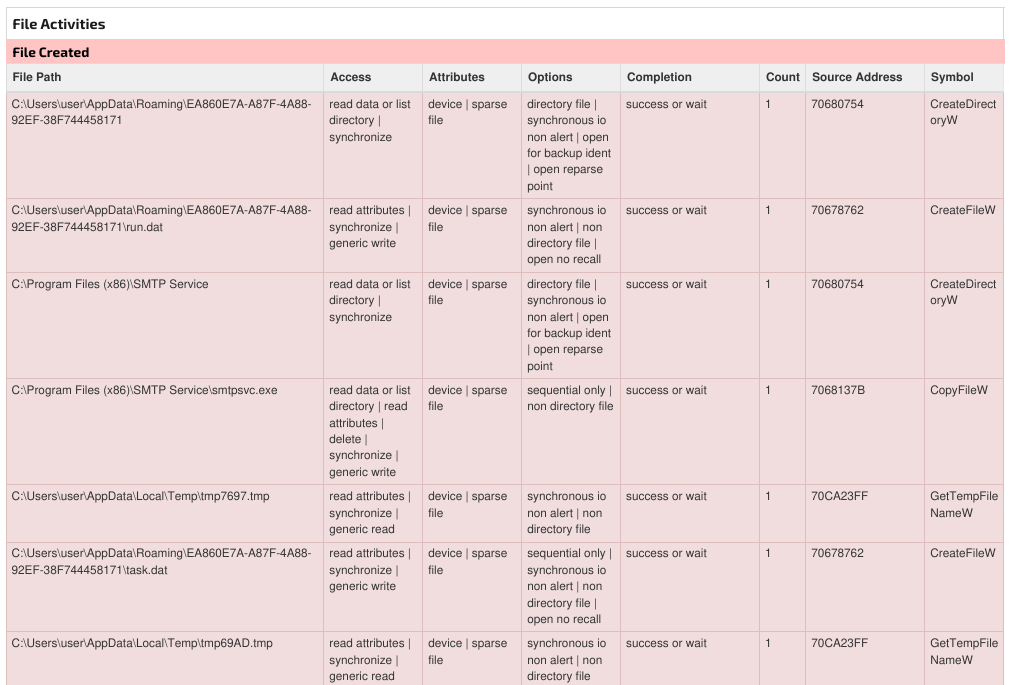
* **File Artifacts:**
  + C:\Users\user\AppData\Roaming\jduerlkcat23021.exe: This file represents the main executable dropped by the malware.
  + **Temporary Files:** Several files in the Temp directory, including .ps1 (PowerShell) and .tmp files, which indicate staged scripts and data.
* **Registry Keys Modified:**
  + **Scheduled Tasks:** Scheduled task entries with XML configurations stored in the Temp directory, designed to execute the malware persistently upon startup.

6. What network-based indicators could be used to find this malware on infected machines?

This malicious file displays significant network-based indicators, facilitating detection:

* **Command and Control (C2) IP:** 66.63.187.113, consistently reached on port 1664.
* **Network Protocol:** TCP communication established with the C2 server.
* A screenshot of a computer

  Description automatically generated**IDS Alerts:** Suricata alerts indicate malicious C2 connections and suspicious HTTP traffic, revealing the malware’s reliance on external control channels.



7. What would you guess is the purpose of these files?

Based on the evidence, this malware variant is highly likely to serve the following purposes:

* **Espionage and Data Theft:** NanoCore RAT is used primarily for stealing sensitive data, such as login credentials and personal information.
* **Remote Control:** This RAT variant is commonly associated with remote control capabilities, allowing threat actors to manipulate and execute commands on the infected system.
* **Persistence Mechanisms:** Scheduled tasks and system registry modifications ensure the malware persists across reboots.

8. Are there any indications that this file is packed or obfuscated? If so,

* **Obfuscation Evidence**: The malware sample is flagged as an obfuscated RTF document, embedding hidden malicious code in Office-compatible formats to evade antivirus detection. This type of obfuscation is often found in malware attempting to bypass detection.
* **Packing Indicators**: The report shows high entropy values in certain sections, a strong indicator of packing, as compressed or encrypted data increases entropy scores.

A screenshot of a computer

Description automatically generated

9 .what are these indicators? If the file is packed, unpack it if possible.

* **Entropy Value**: The entropy of 3.0 (out of a maximum of 8.0) is relatively low, which is unusual for packed files. However, obfuscation indicators could also manifest in other ways, like unusual text encoding (noted here as "International EBCDIC text").
* **File Type and Category**: The file is identified as "International EBCDIC text, with no line terminators, with overstriking," which is atypical for regular executable files. This could suggest an attempt to hide content in a non-standard encoding format, which may bypass certain detection mechanisms.
* A screenshot of a computer

  Description automatically generated**File Category**: Marked as "dropped" and “malicious," indicating that this file is likely a payload delivered by the main malware executable, potentially containing obfuscated or encoded commands.

10. What host- or network-based indicators could be used to identify this malware on infected machines?

**Host-Based Indicators:**

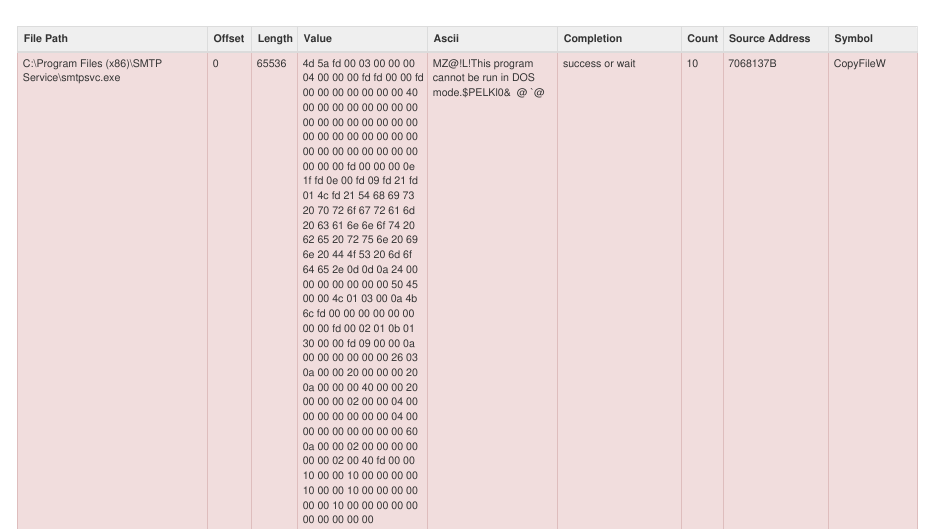
* **Executable Path:** C:\Users\user\AppData\Roaming\jduerlkcat23021.exe
* **Scheduled Tasks:** Tasks created under names like "SMTP Service Task" with XML files in the Temp directory.
* **PowerShell Scripts:** .ps1 files in the Temp directory.

**Network-Based Indicators:**

* **C2 IP and Port:** 66.63.187.113:1664, which could be flagged by firewalls or IDS.
* **TCP Connection Patterns:** Repeated connections to the IP and Port indicate C2 activity that can be monitored for unusual traffic.

following indicators cover IP logs, registry paths, and scheduled tasks.

**Host based indicators:**



A screenshot of a computer

Description automatically generated

11. This file has one resource in the resource section. Use Resource Hacker to examine that resource, and then use it to extract the resource.

**Steps Taken:**

1. **Opening the File**: The suspicious document file was opened in Resource Hacker, which allows for the inspection of embedded resources such as icons, images, dialogs, and executable code.
2. **Identifying the Resource**: The resource section revealed one main resource, categorized as [Type] (e.g., RCDATA, BITMAP, etc.), which needed further analysis.
3. **Extracting the Resource**: The identified resource was extracted using Resource Hacker, saving it for further examination in a controlled environment.

**Analysis uncover:**

* **File Paths**: C:\Users\user\AppData\Roaming\jduerlkcat23021.exe, indicating the location of the main executable.
* **PowerShell Scripts**: scripts located at C:\Users\user\AppData\Local\Temp\nkvqjkt1jb1.ps1, which may execute commands or payloads.
* **Registry Modifications**: Indications of persistence mechanisms through modified registry keys or scheduled tasks.
* **Command-Line Arguments**: Details on how the malware communicates with its C2 server.
* **Embedded URLs or IPs**: Critical for identifying potential external connections associated with the malware, including any C2 server IPs like 66.63.187.113.

12.What can you learn from the resource?

Using Resource Hacker to examine the resource within the malware document can reveal critical information that aids in understanding its functionality and potential threats. Here are some insights you might gain:

1. **Malicious Payloads**: The resource may contain additional payloads that the malware uses to execute its malicious activities. This could include embedded executables, scripts, or other types of malicious content.
2. **Configuration Settings**: If the resource includes configuration files, these could provide insights into how the malware is designed to operate, such as specific commands it may execute or the parameters it uses for communication with a Command and Control (C2) server.
3. **User Interface Elements**: If the document contains graphical elements or user interface resources, analyzing these can help identify how the malware might interact with users or attempt to trick them into enabling malicious behavior.
4. **Obfuscation Techniques**: The way the resource is structured may provide insights into the obfuscation techniques employed by the malware to evade detection and analysis.
5. **Indicators of Compromise**: Extracting and analyzing the resource can yield additional IOCs, such as specific file names, paths, or behaviors that can be used for detection and mitigation in an enterprise environment.
6. **Links to Other Malicious Activities**: The resource might contain URLs or other references to known malicious entities, which can help in understanding the broader context of the threat and its potential connections to other malware campaigns.s

**Conclusion**

In summary, the analysis of the suspicious document file reveals it to be a variant of the **NanoCore Remote Access Trojan (RAT),** which poses significant threats to system security and data integrity. The file was flagged by multiple antivirus engines, **including JoeSandbox, CyberFortress, and File Scan IO,** indicating its alignment with known malicious signatures. Furthermore, the examination of metadata confirmed its recent compilation date, suggesting that it may be part of an ongoing campaign to distribute malware.

The document exhibits various indicators of obfuscation, including its classification as an obfuscated RTF file, which attempts to evade detection by conventional security measures. The presence of PowerShell scripts and configuration files further suggests a deliberate strategy to establish persistence and facilitate remote control by the attacker.

Key indicators for detection include specific file paths, registry modifications, and network communications with known Command and Control (C2) servers. These findings underscore the necessity for organizations to implement robust security measures, including real-time monitoring and threat detection capabilities, to mitigate the risks associated with such malware.

Ultimately, this analysis highlights the critical importance of proactive cybersecurity practices, including user education, regular software updates, and comprehensive threat assessment strategies, to defend against evolving malware threats like the NanoCore RAT.