Skills Learned

- **Static Analysis** (Dissecting binaries without execution)
- **Dynamic Analysis** (Executing malware in a controlled sandbox)
- **Reverse Engineering** (Using tools like IDA Pro, Ghidra, and x64dbg)
- **Network Traffic Analysis** (Using Wireshark and INetSim)
- **Detection Rule Writing** (YARA rules for identifying malware)
- **Process Monitoring & Memory Analysis** (Procmon, Volatility)
- **Virtualization & Network Isolation** (Ensuring malware stays contained)

Tools Used

- **Windows 10 (FlareVM)**: Fully equipped with malware analysis tools.
- **Remnux**: Linux distro for reverse engineering and network analysis.
- **VirtualBox**: Isolated virtual environment for executing malware safely.
- **INetSim**: Simulates internet services to monitor malware behavior.
- **Wireshark**: Capturing and analyzing network traffic.
- **ProcMon & Process Hacker**: Monitoring system behavior.

Static Malware Analysis

Static analysis refers to the process of examining a file, typically malware or a software program, without actually running it.

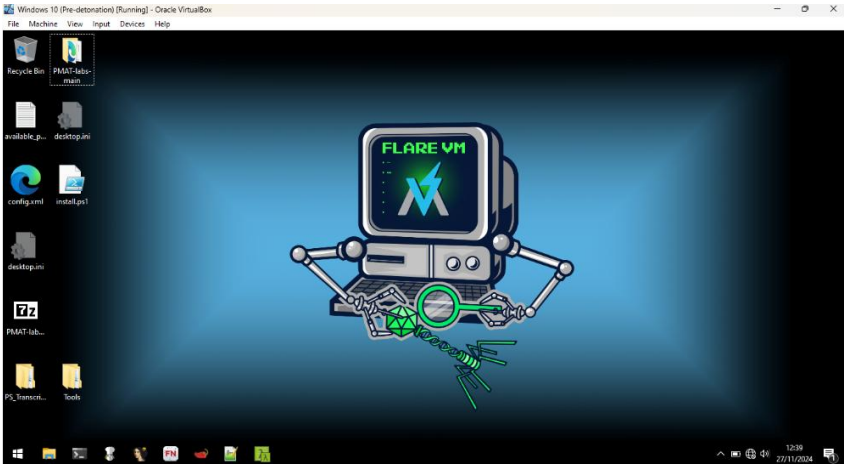
Virtual Environment Setup

- Installed Windows 10 (FlareVM) in VirtualBox with a Host-Only Adapter for complete isolation.
- Installed Remnux in VirtualBox, also using a Host-Only Adapter.
- Ensured the lab is disconnected from the internet to prevent infections from

2 Configuring INetSim for Network Simulation

- Installed and configured INetSim on Remnux.

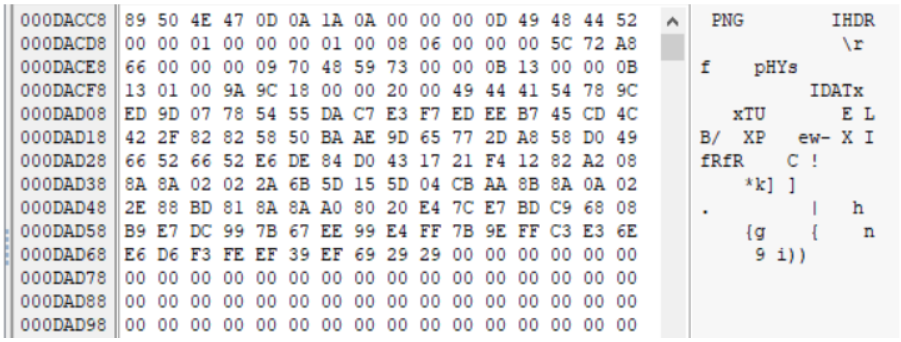
- Set FlareVM to route all traffic through Remnux, allowing malware to "believe" it has internet access.
- Captured DNS, HTTP, and other network interactions with Wireshark.



Identify the file type

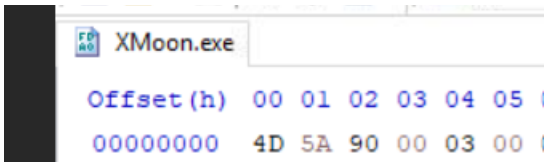
File Type : exe,png,dll

*HxD



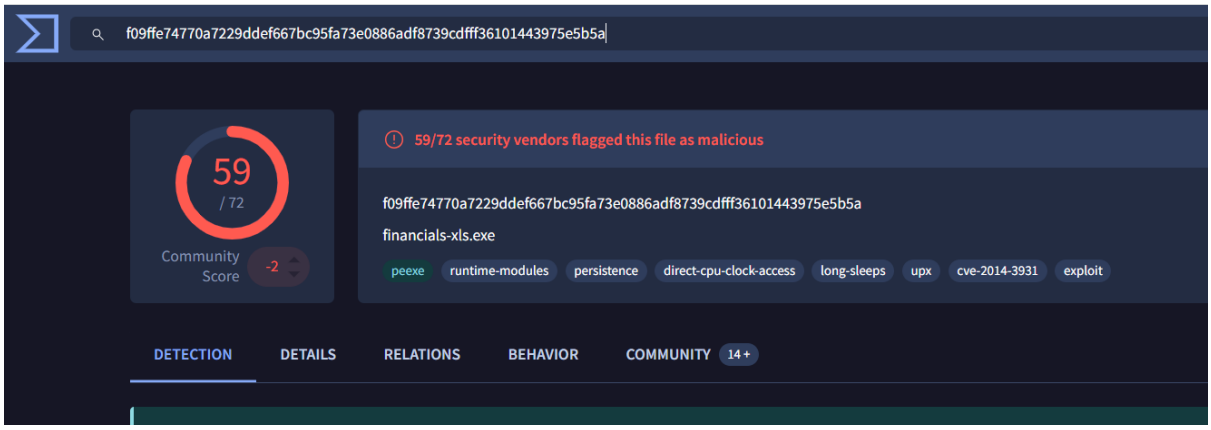
signatures for exes **malware XMoon.exe**

Keywords: MZ, 4D 5A, This program cannot be run in DOS mode

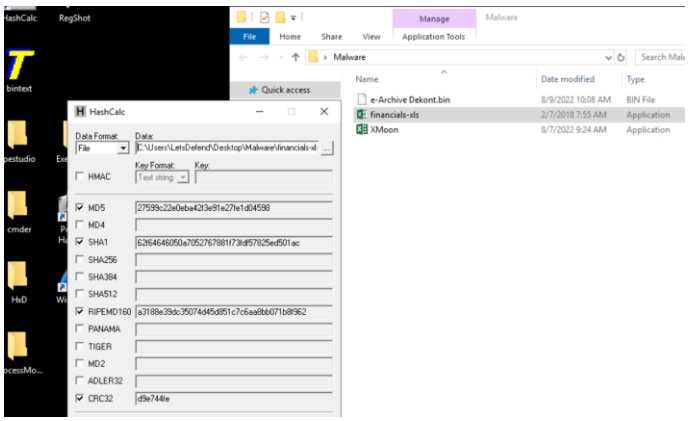


fingerprinting the malware

* Grab the hash and dump it in virustotal



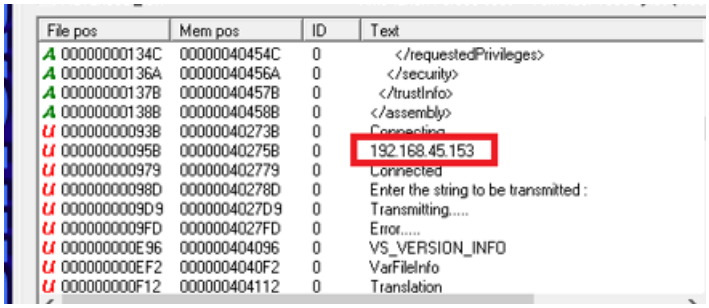
* Hashcal, Hashmyfiles



STRINGS

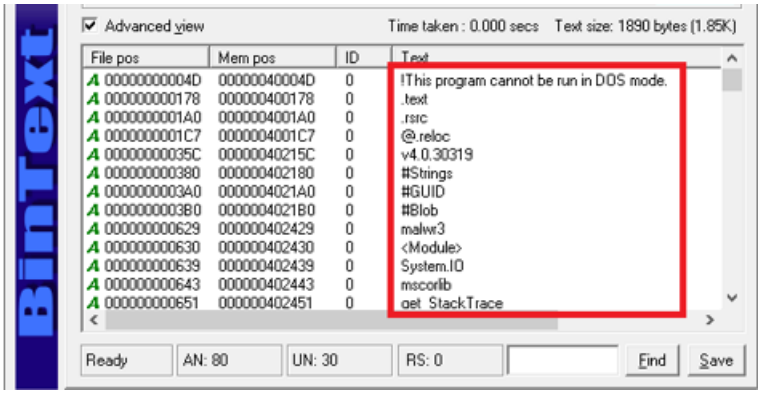
* IP addresses

Base64 or any encoding techniques



Command

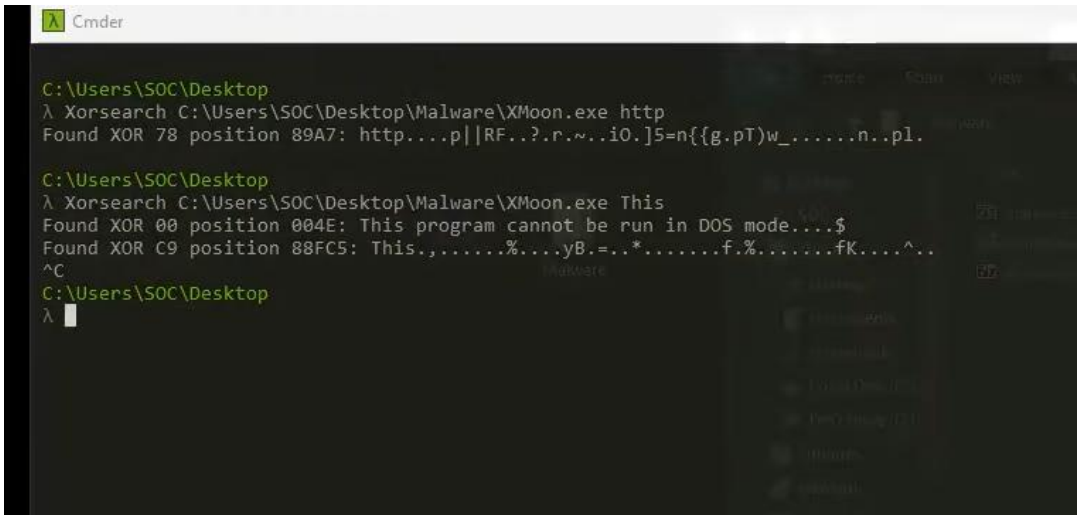
strings



Decrypting Encoded strings

Xorsearch

Commands: **Xorsearch XMoon.exe http**



Automatic Detection

Packing

- **Packing**: The process of transforming an executable file into a different format using a special algorithm.
- **Unpacking**: Reverting the packed file to its original form by reversing the packing procedure.

(Using Packer Detection Tools)

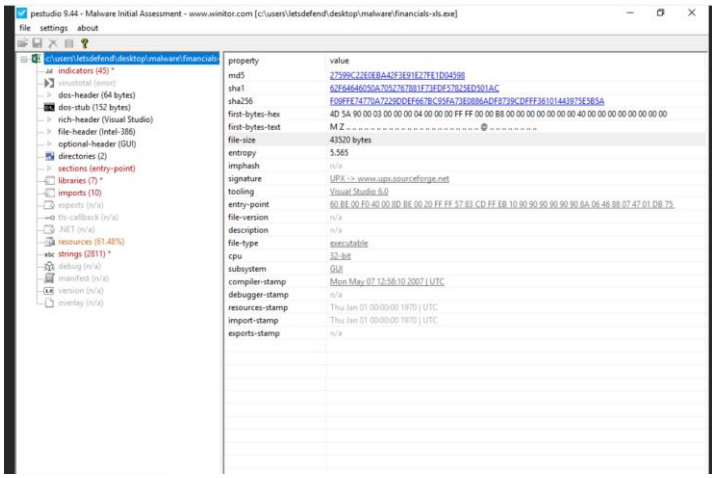
These tools use **signatures** to identify common packers. Examples include:

✓ **DIE (Detect It Easy)** – Identifies packers based on file signatures.

✓ **PEStudio** – Provides metadata and security indicators.

✓ **PEiD** – Detects packers using a database of signatures.

Eg: PEStudio



Checking Section Names in PE Header

Packed executables often have **unusual or extra sections**.

- **Legitimate sections**: .text, .data, .rdata, .rsrc
- **Packed sections**: .upx, .aspack, .themida, .petite

Tools for PE Analysis:

PE-bear

PEView

PE-bear