Project Report

Name: Ethan Conner

Class: COSC-3364-010

Professor: Jo Meier

Topic: Malware Analysis Sandbox

PowerPoint Presentation: MalwareSandboxCybersec.pptx

Project Goals:

- Create a virtual sandbox for safe, isolated static and dynamic analysis of malware.
- Dive into, learn, and use common open-source analysis tools

Steps:

1. Virtual Machine (VM) setup
    a. Selected VirtualBox
    b. OS selection: REMnux, a Linux system for analysis
    c. Configure VM for proper isolation
2. Tool Exploration
    a. Command-line/Forensic tools
        i. Detect It Easy (DIE)
        ii. readelf
        iii. Strings
        iv. Lsof
    b. Static analysis tools
        i. Ghidra
    c. Dynamic analysis tools
        i. Cuckoo sandbox
        ii. Volatility
    d. Network analysis tools
        i. Wireshark

3. Malware Preparation
    a. Download/Transfer/Create live malware
    b. Download/Transfer/Create inactive malware
4. Malware Analysis
    a. Conduct static analysis on malware
    b. Conduct dynamic analysis on malware
    c. Conduct network analysis on malware

Implementation

Virtual Software:

- Oracle VirtualBox
    - OS: REMnux
    - CPU: 4 cores
    - RAM: 8 GB
    - Disk Space: 60 GB
    - Network settings
        - Host only adapter
        - NAT
- These settings are picked knowing modern malware can detect when it is being analyzed, so best to simulate a "real" environment to avoid configuration checks that impact malware's behavior. REMnux is a prepackaged Linux based OS used for analysis. It is preconfigured with lots of libraries and tools used in analysis frequently. I wanted to explore using a preloaded OS, so as to avoid having to deeply configure my Linux environment.

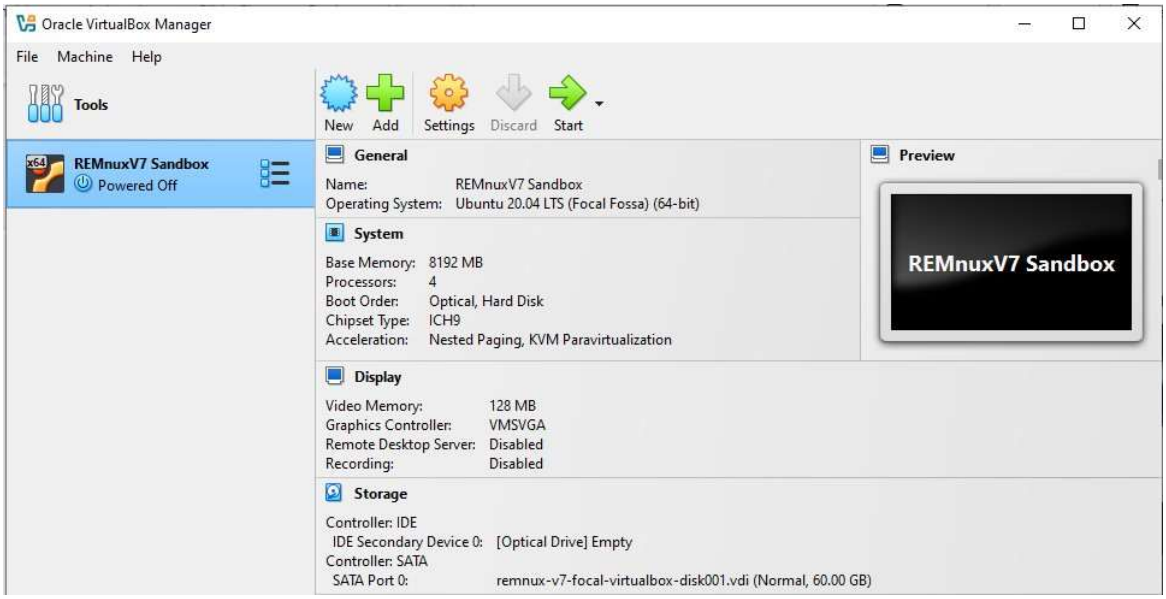

Fig. 1 VirtualBox Configuration

Fig. 2 REMnux OS Logo

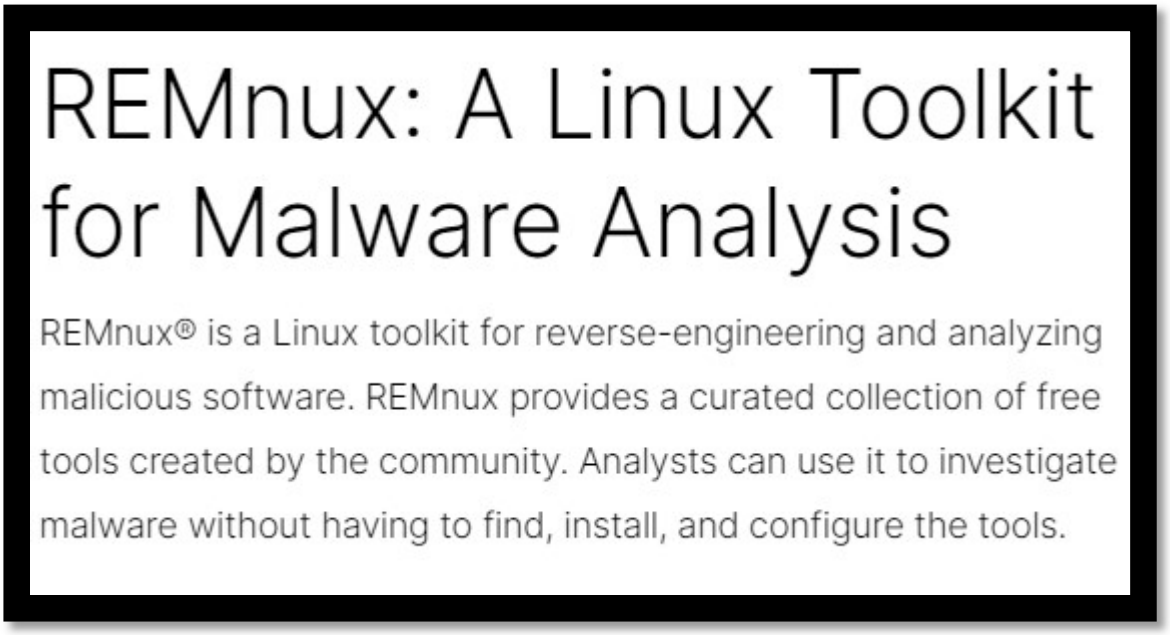

Fig. 3 REMnux homepage

Tools:

- Command-line:
  - readelf
    - A command-line tool that can display detailed information about executable and linkable format (ELF) files, commonly used in Linux. It can provide insights into headers, sections, symbols, objects, and other binary structures that can aid in debugging, analysis, and reverse engineering.
  - strings
    - A command-line tool that extracts and displays readable text from binary files. It is commonly used to identify human-readable content,

such as text or error messages, hidden within executables and other non-text files.

    - losf
        - List Open Files is a command-line tool that shows all open files and the processes accessing them on a system. It is often used for diagnosing resource usage, identifying processes interacting with certain files or network ports, etc. It can be used in dynamic analysis as well.
- Static Analysis:
    - Ghidra
        - A free and open-source software reverse engineering (SRE) framework developed by the NSA. It comprises of a powerful set of tools for analyzing binaries, including decompiling, disassembly, and debugging, across multiple platforms and architectures. It is highly extensible, allowing users to develop custom plugins and scripts for specific analysis needs. It is widely used in malware analysis and vulnerability research. You can get decompiled code from payloads without any execution and even determine functions, investigate data structures, and uncover obfuscated code or encrypted data.
- Dynamic Analysis:
    - Cuckoo Sandbox
        - An open-source automated malware analysis tool that performs dynamic analysis by operating at the hypervisor level to execute malware in a logged, isolated environment. It observes and records behavior of the OS, file system, network activity, registry, and process creation. It then generates detailed reports on actions taken by the malware, allowing insight into functionality and potential impact of malware. It is an extremely comprehensive way to detect and study malware in a controlled environment.
    - Volatility
        - An open-source memory forensics framework used to analyze volatile memory (RAM) dumps for digital information. It can extract detailed information about running processes, network connections, in-memory data structures, and more to identify malicious activity or retrieve sensitive data. It supports a wide range of memory dump formats and OSs.
- Network Analysis
    - Wireshark

- An open-source network packet analyzer that captures and inspects data packets traveling across a network. You can view detailed information about network traffic, including headers, payloads, protocols, and more making it a very powerful tool for performance monitoring and security analysis. It supports a wide range of protocols and can be used to detect security vulnerabilities or analyze suspicious network activity. It has extensive filtering and analysis capabilities making it a must learn tool for security professionals.

*Note: There are many other network tools, however due to complexity and extensiveness of Wireshark, I am not really exploring other tools in network analysis yet.

Malware:

- VM is not ready for live malware yet, having SSL certification issues, so something is wrong with authentication on my VM for downloading things through web transport. This means I am potentially having network errors.
- Alternative: Created my own Metasploit script using msfvenom to statically analyze.

- https://docs.metasploit.com/docs/using-metasploit/basics/how-to-use-msfvenom.html
- https://www.offsec.com/metasploit-unleashed/msfvenom/

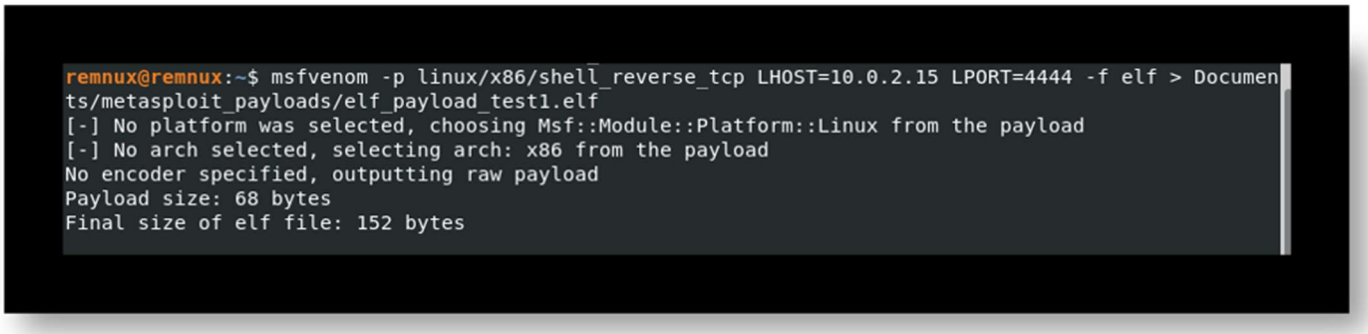

Fig. 4 Metasploit payload creation

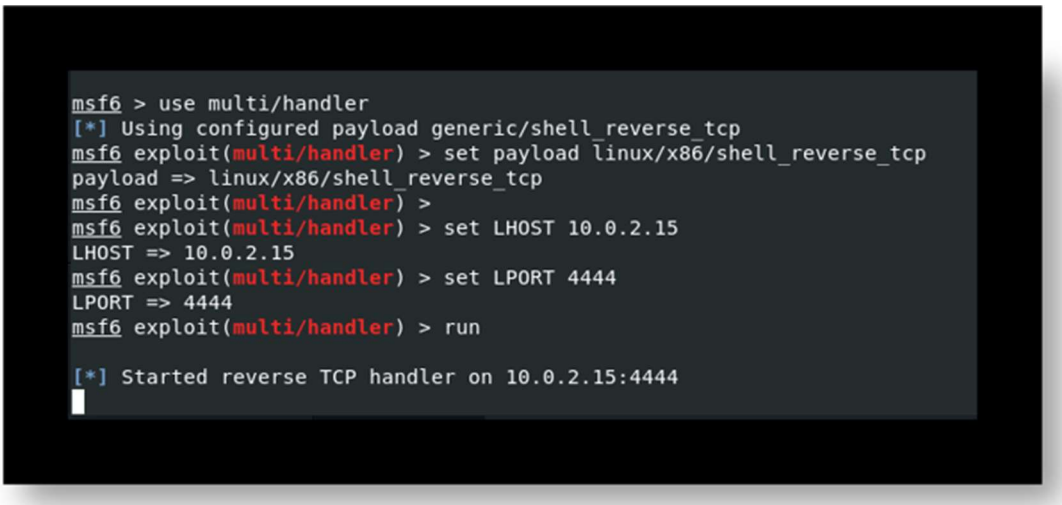

Fig. 5 Setting payload on own VM

Malware Sources:

- TheZoo: https://github.com/ytisf/theZoo
  - Live malware repository on GitHub
- MalwareBazaar Database: https://bazaar.abuse.ch/

Analysis of Payload:

- Snapshots
  - A backup or saved state of the VM at a specific moment in time
  - Saves information such as:
    - OS
    - Running applications
    - Configuration
    - Data
  - Useful for resetting environment after destroying them
  - Quick recovery

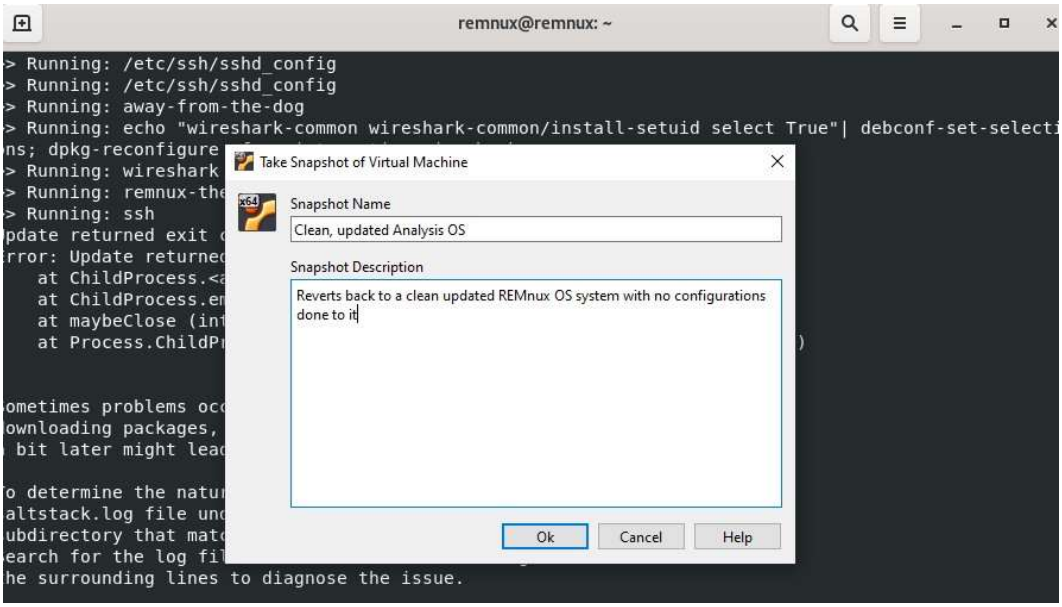

Fig. 6 Taking snapshot of machine

- readelf
  - Gathering some header data from the exploit payload

```
remnux@remnux:~$ readelf -h Documents/metasploit_payloads/elf_payload_test1.elf
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Intel 80386
  Version:                           0x1
  Entry point address:               0x8048054
  Start of program headers:          52 (bytes into file)
  Start of section headers:          0 (bytes into file)
  Flags:                             0x0
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         1
  Size of section headers:           0 (bytes)
  Number of section headers:         0
  Section header string table index: 0
remnux@remnux:~$
```

Fig. 7 readelf on the payload

```
Program Headers:
  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
  LOAD           0x000000 0x08048000 0x08048000 0x00098 0x000dc RWE 0x1000
```

Fig. 8 Program headers found with readelf

- Detect It Easy (DIE)
  - Used to look at payloads and assembly language

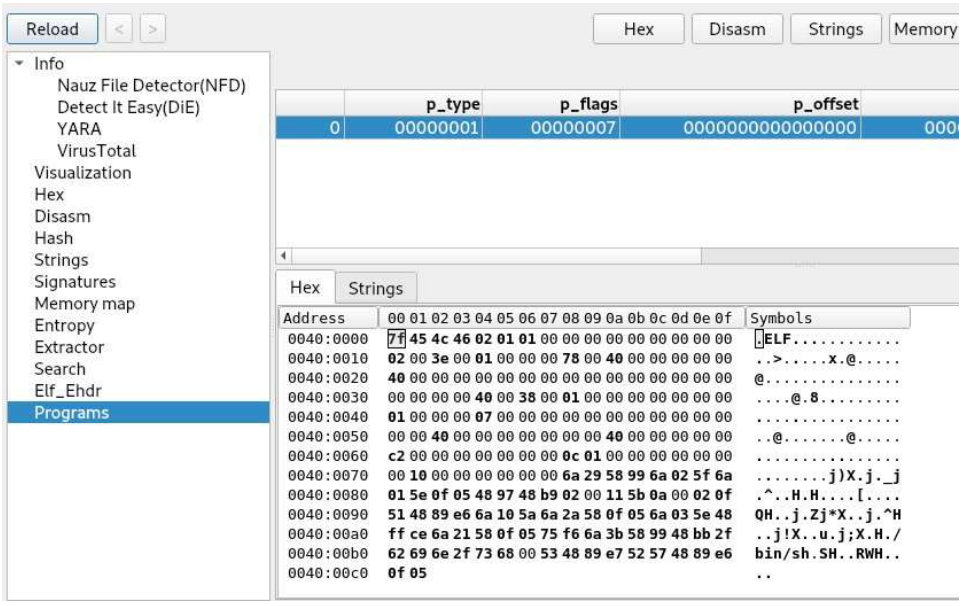

Fig. 9 DIE interface

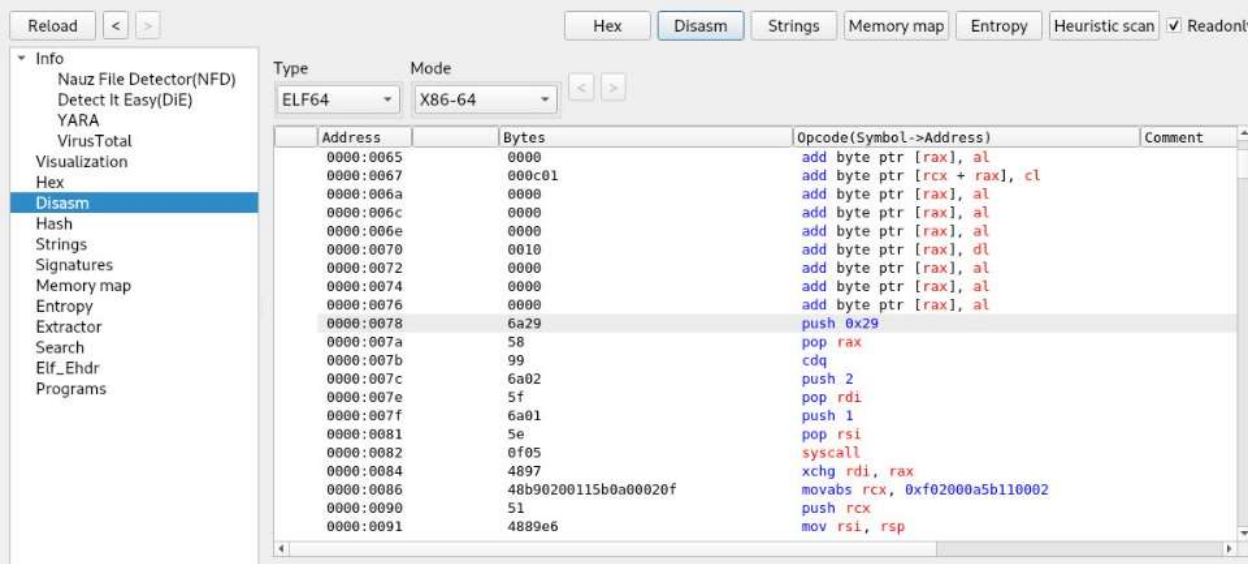

Fig. 10 Disassembly instructions

- Ghidra
  - Used for static analysis of payload, to reverse engineer it and gather information about it. Can also view functions.

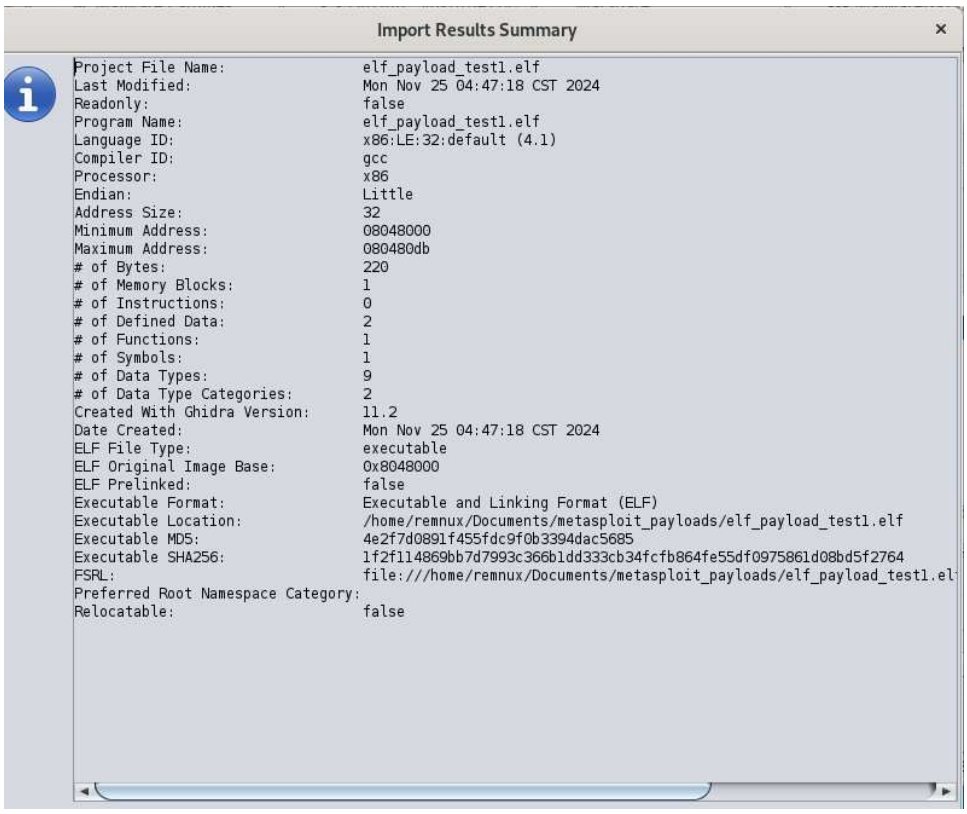

Fig. 11 Ghidra summary results page.

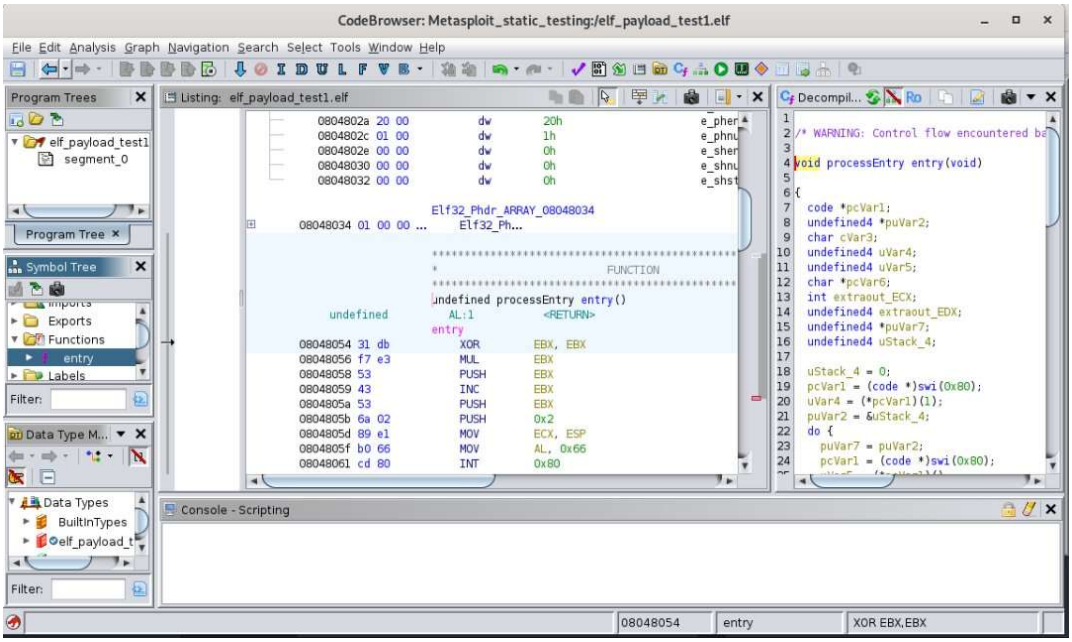

Fig. 12 Ghidra, decompiled code from payload on right tab

Next steps:

- Fixing SSL certification issues/sort out network issues
- Either get second PC or backup current PC
- Set up multiple VMs, one for malware execution, one for analysis
- Download live malware once safe to do some dynamic analysis on with Wireshark and Cuckoo Sandbox

Documentations:

- Virtual Machine
  - https://www.virtualbox.org/manual/
  - https://www.virtualbox.org/wiki/Technical_documentation
- OS
  - https://remnux.org/#docs
  - https://docs.remnux.org/
- Forensic tools
  - readelf
    - https://www.man7.org/linux/man-pages/man1/readelf.1.html
  - strings
    - https://man7.org/linux/man-pages/man1/strings.1.html

- DIE
  - https://github.com/horsicq/Detect-It-Easy
- lsof
  - https://man7.org/linux/man-pages/man8/lsof.8.html
- Ghidra
  - https://github.com/NationalSecurityAgency/ghidra
- Cuckoo Sandbox
  - https://cuckoo.readthedocs.io/en/latest/
- Volatility
  - https://volatility3.readthedocs.io/en/latest/
- Wireshark
  - https://www.wireshark.org/docs/wsug_html_chunked/
  - https://www.wireshark.org/docs/

Sources (videos and webpages):

- https://www.youtube.com/watch?v=_80fpeY-_AI
- https://www.youtube.com/watch?v=BluBQd9-Fc8
- https://www.youtube.com/watch?v=9TEeribDUXE
- https://www.youtube.com/watch?v=as6qWIV8IWM
- https://www.youtube.com/watch?v=1lKPeKkWvR8
- https://www.youtube.com/watch?v=oPsxy9JF8FM
- https://www.youtube.com/playlist?list=PLR_k_vG4Lz0FaCRrbWf5pWlS6Q9Hnw9On