**Lab 6**

**Firmware**

Clayton Davidson | 000860643

April 22, 2022

**Part 1**

1. The root password is: $1$Jl7H1VOG$Wgw2F/C.nLNTC.4pwDa4H1:18145

Text

Description automatically generated

1. The username of user #1000 is : iotgoatuser

Text

Description automatically generated

1. The architecture of this system is: i386 - ELF 32-bit

A screenshot of a computer

Description automatically generated

I had some conflicting sources, emba identified the file as x86; binwalk ARM 16bit.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated

1. The IP address of LAN is: No IP address specified

Text

Description automatically generated

1. Entropy Analysis
   * The firmware appears to be compressed due to very consistent levels of entropy through the file.
   * The compression is xz

Graphical user interface, text

Description automatically generated

* + This file appears to be firmware meant for embedded systems

Graphical user interface

Description automatically generated

**File System**

The first interesting thing I found was a file “s95\_interesting\_binaries\_check.txt “ that logged four binary files for post-exploitation.

Text

Description automatically generated

The second interesting thing I noticed was the lack of any certificates within the firmware binary. This, combined with the tagged post-exploitation binaries and the high level of file entropy, could be indicators of a malicious firmware file.

Graphical user interface, text

Description automatically generated

**Ghidra**

Text

Description automatically generated

Ghidra would not analyze the Firmware sample, but something that would be nice to find in Ghidra is any strings that might hint at functionality, or connection credentials that may be hard coded into the firmware from the manufacturer.

**Critical Reflection**

This was an eye-opening look into real firmware sample analysis. I was surprised to learn that the file would have its own filesystem like other Linux file structures. It never occurred to me that it would have a specific file structure similar to an Operating System, simply that there would be a single large program that produced all the system functionality. Knowing the file structure and the code language of the file allows for much more in-depth analysis with tools such as Ghidra and emba. Binwalk is also a valuable resource as it provides preliminary analysis functionality.

This is an excellent example of the importance of security on embedded/IoT devices. If this firmware was retrieved from a device stolen by a potential attacker, reverse engineering the firmware could reveal critical system and authentication data as well as allow remote access for an attacker.

Ghidra was the largest difficulty I faced, there was an error in Java when attempting to analyze with the “ARM Aggressive Instruction Finder” selection. I used the information provided in your videos to infer what valuable information may be found through Ghidra. I also had the “np” issue with binwalk, as well as a problem loading the dependency “sasquatch” which I was able to resolve.

I was uncertain as to which Architecture was correct for this firmware sample, as I found 3 different evaluations. I also could not find any IP address assigned to the “LAN” connection. Could you clarify how to find the IP on this sample and which architecture is correct?

I feel that I have learned a lot about firmware and IoT/embedded devices, but the biggest revelation was that the filesystem structure would resemble a Linux Operating System filesystem.