Lab Template – Ethan Roepke

1. **Identify two additional architectures for which the Mirai dropper has been compiled. Don't give me just the abbreviations. You must give me the full names. Just because a device has this architecture, doesn't mean it is susceptible to Mirai.**
   (10 points)

   Bins/dlr.arm – binaries dynamic language runtime
   Bins/dlr.h4 – refers to the super H 4 processor

2. **What service would need to be enabled to be susceptible to Mirai?**
   (10 points)

   Telnet

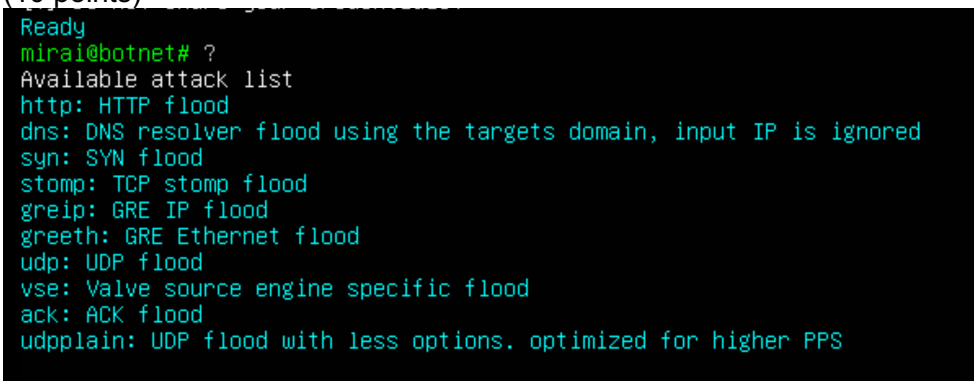
3. **Explain the line containing three commands. What are the commands doing?**
   (15 points)

   /bin/busybox wget… - Busybox is command calls the busybox executable and wget with the following URL retrieves the file mirai.x86 and drvHelper will save the output into a file.

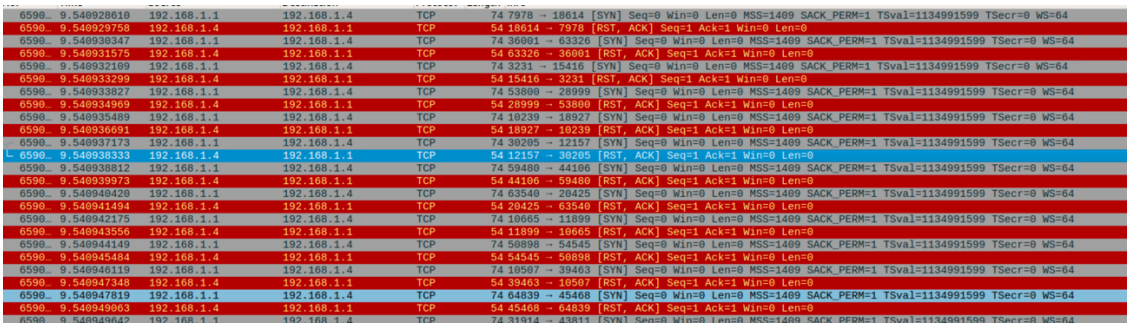
   /bin/busybox chmod… - Busybox command is called executable. Chmod will modify permission, given 777 therefor everyone will have r-w-e. And lastly drvHelper is the file they are referring to for the permission modifications.

   /bin/busybox ECCHI – Busybox command calls the busybox executable and ECCHI is not a part of the standard busybox and a separate configurations so I am not sure what the role of it is used for.

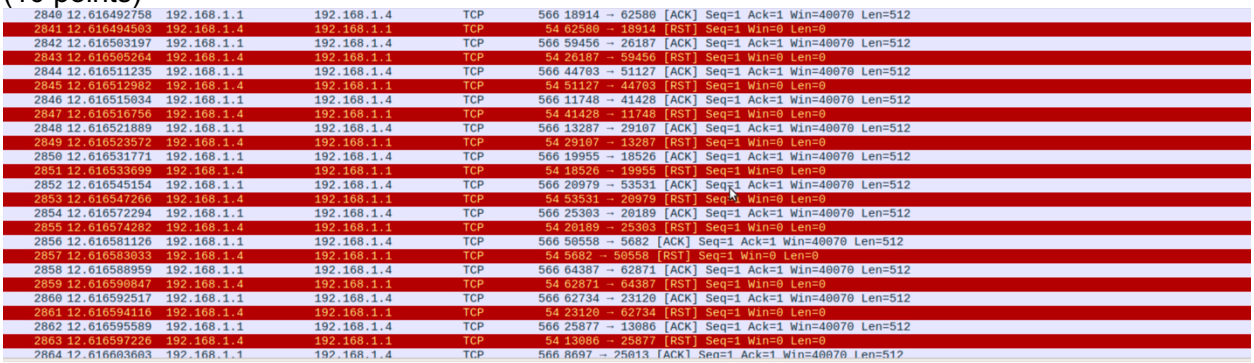
4. **Screenshot of help (?) from the CNC command line.**
   (10 points)



5. **Screenshot of wireshark of successful syn flood attack**
   (10 points)



6. Screenshot of wireshark of any other successful attack from Mirai's list of options (10 points)



7. What types of IoT devices (Fridges, Microwaves, etc…) were primarily exploited in Mirai attacks? You may refer to this link.

(10 points)

After reading the file, the most common compromised devices from the article were cameras and digital video recorders.

8. What are some of the commonalities found in the devices in your previous answer? (Think about OS, versions, services…)

(10 points)

These two devices that became compromised were running a BusyBox on linux so Mirai are able exploit these devices vulnerabilities.

9. How did the developers of Mirai ensure that they didn't attack private IP addresses, the USPS, nor the DOD? Why would they want to do this?

(15 points)

The developers developed a filter which would have the bots programmed to avoid specific IP addresses that they need to avoid. Doing this would decrease attention and avoid consequences from illegal activity. Being able to eliminate Private IP addresses would enhance the efficiency of the process because they are non-routable and inaccessible from external networks.