JOHNS HOPKINS WHITING SCHOOL of ENGINEERING | Information Security Institute

EN.650.624, Network Security, Spring 2020

Instructor: Reuben Johnston, reub@jhu.edu

Homework 2

See Blackboard for grade %

Due Date

See Blackboard for date

Summary

Network application layer exploration with a raspberry pi.

- Team 1-email server (postfix, Dovecot, MariaDB mySQL) on rpi
- Team 2-http server (nginx) on rpi and proxy server on rpi (privoxy)
- Team 3-media server (kodi UPnP/Server) on rpi
- Team 4-ftp server (pure-ftpd) on rpi, nfs, and smb/cifs server on rpi (samba)
- Team 5-dhcp server (isc's dhcpd) on rpi and dns server (isc's bind9) on rpi
- Team 6-game server (minecraft) on rpi
- Team 7-p2p server (BitTorrent) on rpi

Part 0

Prepare your build host and client target: Kali on x86_64 VMware guest with Docker. Prepare your server target: Ubuntu server on RPI4B host with Docker.

Part 1

KM preparations – create your GitHub wiki markdown file. The file should eventually summarize all the steps performed in the project, and along with the Dockerfiles, enable another student to setup and use these applications easily. Locate the source code for the application server and client and include links for it in the markdown file. We will import this into our NwSec wiki, along with the Dockerfiles described below.

Part 2a – server build environment

Prepare your Docker container for building RPI4B binaries (application server) using the build host. For the final form of the container, create a Dockerfile that will replicate its setup. Build the appropriate form of the application server code in the container, and include instructions to replicate in the documentation.

Part 2b – client build environment

Prepare your Docker container for building host binaries (application client) using the build host. For the final form of the container, create a Dockerfile that will replicate its setup. Build the appropriate form of the application client code in the container, and include instructions to replicate in the documentation.

Part 3a – server runtime environment

Prepare your target server's runtime environment: application server running in Docker container on RPI4b hardware running Ubuntu Server. Create a Docker container for the application server's runtime environment. For the final form of the container, create a Dockerfile that will replicate it and include instructions on how to setup and run the server application in the container.

EN.650.624, Network Security, Spring 2020 Instructor: Reuben Johnston, reub@jhu.edu

Part 3b – client runtime environment

Prepare your target client's runtime environment: application client running in Docker container on x86_64 VMware guest running Kali. Create a Docker container for the application client's runtime environment. For the final form of the container, create a Dockerfile that will replicate it and include instructions on how to setup and run the application client in the container.

Part 4

Perform network capture of traffic between demo host running the application client and target running the application server.

Requirements

Executive summary report

The report must contain an overview of the project, the project goals, project execution highlights (i.e., what was performed, with pertinent technical highlights), what goals were accomplished (and any that were not achieved), and what was learned from the project (2-3 pages). For the report format, groups shall use the IEEE format (see here). The report is intended to be reference material for when you are performing threat modeling in practice.

Groups will begin the project with the creation of a project backlog (see Lecture 2 for reference) and by assigning tasks in the backlog amongst the team. For each task in the backlog, groups should include exit conditions and schedule milestones.

For this assignment, place supplementary code and documentation in the specified locations on the course GitHub. Submit supplemental *.pcap files in a single *.zip archive using BB. Turn in the executive summary report, via TurnItIn assignment on BB, and your supplemental *.zip in the separate BB assignment for supplemental material. Please include team member last names in the filenames for all files submitted (e.g., "hw2-execsum-name1-name2-name3.docx" or hw2-supplemental-name1-name2-name3.zip).

Deliverables

Part 1

1. Provide markdown file describing requested details for parts 2-4; include sections in the markdown file for each of parts 2-4.

Part 2a

2. Provide a Dockerfile for replicating the application server build environment and include instructions for using it to retrieve and build the source code.

Part 2b

3. Provide a Dockerfile for replicating the application client build environment and include instructions for using it to retrieve and build the source code.

Part 3a

4. Provide a Dockerfile for replicating the application server runtime environment and include instructions for using it to run the server application.

EN.650.624, Network Security, Spring 2020 Instructor: Reuben Johnston, reub@jhu.edu
Part 3b

5. Provide a Dockerfile for replicating the application client runtime environment and include instructions for using it to run the client application.

Part 4

6. Provide *.pcap network capture file from traffic between demo host running the application client and target running the application server and describe the sequence used.

Grading Rubric

Executive summary report (50% overall)

Table 0.1-Executive summary grading rubric (2006, Leydens, Santi)

Objective	1 - Exemplary	2 - Proficient	3 - Apprentice
Format / layout / organization	Report is very clear, coherent with excellent transitions	Report is clear and coherent, strong throughout	Report has some gaps, some weak sections
Writing mechanics	Report is virtually error-free, and contains few if any reader distractions	Report is logical and easy to read, and may contain a few errors causing minimal reader distraction	Report is generally clear, but distracting errors and flow make it difficult to follow at times
Figures / Tables	All figures and tables are easy to understand, and are clearly linked to the text. Story can be told almost entirely through figures.	All figures and tables can be understood with information given and are linked to text. One or more need improvement. May need more figures to tell the story.	Figures and/or tables are hard to understand, are not all linked to text. Several need improvement. Several more figures are needed to tell story.
References	All sources identified and referenced appropriately. Evidence of careful and thorough research for outside information.	All sources identified and referenced appropriately. Includes mostly readily available works.	All sources identified. Only readily-available works included. Some weaknesses in referencing, such as missing publisher information.
Typical Grade (average):	90-100%	80-90%	70-80%

Threat modeling deliverables (50% overall)

- Markdown documentation – 25%
- Build environment Dockerfiles and instructions – 15% and 10%
- Runtime environment Dockerfiles and instructions – 15% and 10%

- *.pcap network capture file and description of sequence – 15% and 10%

References and useful resources

- https://github.com/jhu-information-security-institute/NwSec/wiki
- 2006, Leydens, Santi, “Optimizing Faculty Use of Writing as a Learning Tool in Geoscience Education” (provided the rubric specified for the executive summary report)