Lab1Milestone1

Jump to bottom

sethnielson edited this page on 8 Oct · 2 revisions

Lab 1 Milestone 1: Handshake

Assigned	10/7/2019
Due	10/14/2019
Points	75

Overview

It's time to get started creating a "session" layer for Playground.

There are a lot of to-do items for this lab, so please pay attention.

Creating a Playground Network Layer

In real life, putting in a network protocol lower than the application layer requires a device driver. Playground allows you to insert a protocol stack between the wire protocol and the application protocol using a "connector." The basic instructions for inserting a connector can be found here.

Naming the Lab 1 Protocol

As explained in the document, you need to name protocols. We won't call ours TCP/IP, but what we do call it is up to you. The PETF should name the protocol by the due date.

Previous semesters have used the names such as PCP (Playground Control Protocol) and PIMP (I can't even remember what it stands for).

So, PETF, by 10/14/2019, on slack, you must NAME the lab1 protocol.

The Handshake

Each team needs to come up with a "handshake" that is used by the lab 1 protocol to create a session, similar to TCP's 3-way handshake. You do not need to follow TCP's model in any way, but you must get the same properties. Specifically:

1. The handshake must establish a *new* connection (that is, you can tell one connection from another)
2. Both sides must be aware of the other before the connection is "established"
3. The server must have some mechanism for telling the client that the connection cannot be established (e.g., that there was an error or otherwise something went wrong).
4. While you haven't yet figured out how you're going to do the reliable delivery, any setup information must be communicated as part of the handshake. You can modify the handshake later as needed for subsequent parts of this lab.

In practice, you need to figure out how to have your lab 1 layer establish the connection first by transmitting whatever packets are needed between itself and its peer. Only once the connection is established (the handshake is complete) should the higher layer know that the connection is made. Please think about how to do this in the code before asking for help. I'd like you to at least try and figure it out.

Once the higher layer (the application layer) can connect, you can provide it a "pass through" transport. For now, just allow communications to travel directly through without modification or processing.

I am not requiring shutdown messages at this part of the lab, but you're welcome to experiment with this concept if you'd like.

Writing the PRFC and Standardizing by the Due Date

Each team will be taking their own approach to this. At any time you can start to propose to other teams how you think the class should do the assignment. But you will be more persuasive if you have a working prototype. As you get your lab working, you need to discuss over slack how you think the PRFC should be written and what should be the standard. Unlike the Escape Room exercise, this will involve protocol design and not just packets.

The PETF can vote, at any time, on any proposal. BUT the handshake must be standardized and the whole class conforming to it by the due date (10/14/2019).

Grading

On the due date, I will post an auto-grader on the SAFE NETWORK that conforms to the proposed PRFC. I will provide you with the auto-grade client and you will not need to write it yourself. But it will require that the connector for your network layer is installed in your own personal playground with the name assigned by the PETF.

The address and port of the autograder will be:

```
20194.0.0.19100 and port 19101
```

The auto-grader will test handshakes in both directions. You will be graded as follows

- 25 points for correct client-side three-way handshake
- 25 points for correct server-side three-way handshake
- 25 points for correct handling of dropped or corrupted handshake packets

▼ Pages 26

Find a Page...

Home

BackgroundOverlayNetworks

Creating an unreliable switch

Exercise10MonitorPlayground

Exercise1GettingStarted

Exercise2EscapeRoomSockets

Exercise3EscapeRoomAsyncio

Exercise4EscapeRoomAsychUserInput

Exercise5EscapeRoomPlayground

Exercise6EscapeRoomPackets

Clone this wiki locally

```
https://github.com/CrimsonVista/20194NetworkSecurity.wiki.git
```