Lab2Milestone1

Jump to bottom

Karan Dhareshwar edited this page 7 days ago · 3 revisions

Lab 2 Milestone 1: Cryptographic Handshake

Assigned	11/7/2019
Due	11/14/2019
Points	100

Overview

Building on top of our reliable layer (poop), we're going to create a secure layer similar to TLS or Kerberos.

Unfortunately, given that so many teams have slackers, we are going to do this assignment partially in groups and partially individually. Each individual will make an independent submission.

Setting up the Stack

In the previous lab, you set up a connector in Playground for Poop. This time, you will create a two-layer stack. You will need poop included, even if you have a separate individual poop layer. In your module's `__init__.py` instead of creating a Stacking Factory with one protocol, it will have two:

```
SecureClientFactory = StackingProtocolFactory.CreateFactoryType(poopClient,
secureClient)
SecureServerFactory = StackingProtocolFactory.CreateFactoryType(poopServer,
secureServer)

secureConnector = playground.Connector(protocolStack=(
    SecureClientFactory(),
    SecureServerFactory()))
```

Creating a Design and A Name

This part can be done as a team. Decide if you want a TLS or Kerberos style protocol. For this phase, you only need to define the handshake that establishes the cryptographic keys necessary for a secure channel.

Implement the Handshake

This must be done individually. Based on your team's design, each member of the team implements the protocol. You SHOULD test with each other to ensure compatibility and correctness. You can even share unit tests with each other.

Propose a PRFC draft

The team collectively must create the draft to propose to other teams of the PRFC. Each team still has one PRFC member.

Handshake Requirements

At the end of this phase, you should have a secure layer on top of the reliable layer that performs a cryptographic handshake. After the handshake, allow data to be transported unencrypted. Performing the encryption will be done in the next lab.

By the end of the handshake both parties should have the keys necessary for a secure tunnel (confidentiality and data origin authentication). Unlike regular TLS, your protocol should send a certificate in both directions and perform "mutual" authentication. For more info, search about "mTLS" or "mutual TLS".

This is important: you do NOT have to do the full authentication on the certificate at this stage. For now, simply assume that the certificate transmitted is legitimate. In milestone 2, you will validate the certificate.

Grading

There will be an autograder. More details forthcoming, but it will be similar to previous autograders. This one will check to see that, at the end, it has the necessary keys.

Get the autograder files in the repo, and run the client file as follows `python autograder_lab2_client.py 20194.0.0.19000 <team_number> milestone1 submit`

NOTES: The autograder is running on the reliable class switch and is on port 19102

▼ Pages 26

Find a Page...

Home

BackgroundOverlayNetworks

Creating an unreliable switch

Exercise10MonitorPlayground

Exercise1GettingStarted

Exercise2EscapeRoomSockets

Exercise3EscapeRoomAsyncio

Exercise4EscapeRoomAsychUserInput

Exercise5EscapeRoomPlayground

Exercise6EscapeRoomPackets

Exercise7EscapeRoomAdmission

Exercise9StandardizeEscapeRoom

Lab1Milestone1

Lab1Milestone2

Lab1Milestone3

Show 11 more pages...

Clone this wiki locally

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