ISP Programming Assignments

CS GY 6813 Information Security & Privacy

Professors:
Paola Garcia, MS.
Brandon Sloane, MS.
Aspen Olmsted, Ph. D.

9.12.2021

Programming Assignments – Overview

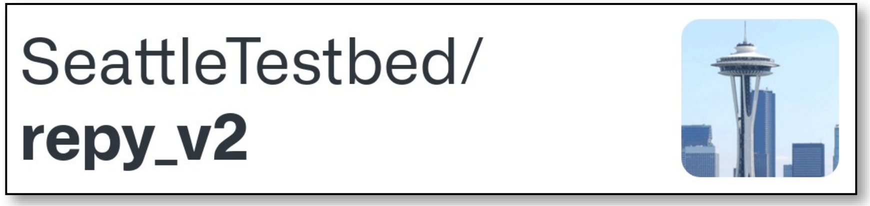
Breadth

IMMERSIVELABS

A collection of Cybersecurity challenges which allow you to play around with and learn from a variety of different domains.

- Each lab has a learning element and then some practical application.

Depth



SeattleTestbed/
repy_v2

A Python programming assignment that digs into how a reference monitor (defines read/write operations) works.

- First, you will take an insecure reference monitor and figure out how to make it secure.
- Second, you will write a series of attacks against our insecure reference monitors to demonstrate how they can be exploited.

Programming Assignments – ImmersiveLabs

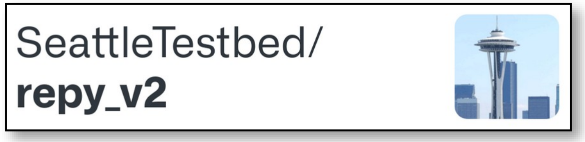
1) Create an account on https://dca.immersivelabs.online (*You MUST use your NYU.edu email address*)
2) Choose any persona you'd like (it won't matter)
3) Navigate to each of the Labs for the week
4) Complete the Lab
5) Capture a screenshot showing you completed it

How to submit:

Upload the screenshot to Brightspace under "Assignments"

Lesson 1	• Cyber Essentials • Compliance, Legislation, Regulation and Standards
Lesson 3	• Immersive Bank – Episode One: Open Source and Credentials • Immersive Bank – Episode Two: Gaining Access
Lesson 4	• Policy, Process and Procedure • GDPR Aware
Lesson 5	• Getting hashed • Hashing – MD5 • Hashing – SHA-1
Lesson 6	• John The Ripper • Multi-Factor Authentication • Password Hashes II
Lesson 7	• Analysing Sandbox Reports
Lesson 8	• Command Line Introduction • Moving Around • Linux File Permissions
Lesson 9	• Software as a Service (SaaS) • Containers • Infrastructure as a Service (IaaS) • Platform as a Service (PaaS) • Virtualization
Lesson 10	• SQL: An Introduction • SQL Injection: UNION • Cross-Site Scripting (XSS) – Reflected
Lesson 11	• Domain Names • Tor and Tor Hidden Services
Lesson 12	• Cryptocurrency & Blockchain

Programming Assignments – RePy



SeattleTestbed/
repy_v2

Setup the Environment:

We have supplied a Virtual Machine with RePy full installed and ready to go.
If you prefer to do-it-yourself, there are build instructions on github.

Part 1:

We have provided an insecure reference monitor (set of rules for how to handle reads, writes, etc.). Your job is to **modify the code to make it more secure**. Think about different ways that attackers might compromise the reference monitor and then things you can do to prevent that from occurring. Link to instructions: Programming Lab Part 1

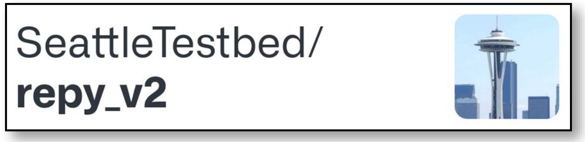
Part 2

Now you are the attacker. We have created a series of insecure reference monitors and your job is to **create a series of attacks to compromise our reference monitors**. You could theoretically accomplish this with a single attack that works across all of our monitors but you will likely be much more successful writing a handful of different attacks, that each try to do something different. Link to instructions: Programming Lab Part 2

How to submit:

- Go to Gradescope and submit your files following the naming conventions as below:
  - For Programming Lab Part 1: reference_monitor_[**netid**].r2py
  - For Programming Lab Part 2: attackcase[**#**].r2py (multiple files submissions are allowed, or you can combine your attacks into one file)
  Note: Your file needs to be all letters and in lower case
- Do not raise any errors or produce any output

Programming Assignments – RePy



SeattleTestbed/
repy_v2

Frequently Asked Questions:

- **Q:** Do I have to turn in my code to Brightspace?
  A: No. You only need to submit to Gradescope. We will import scores on a by-weekly basis and there will be a final grade import run at the end of the semester.

- **Q:** Why am I not getting full points?
  A: The autograder test cases will give you rough hints about which tests you are failing. If you are still struggling, try reaching out to one of the TA's or attend one of the office hours.

- **Q:** Why am I getting a 0/10 even though I know I'm passing some of the test cases?
  A: If your code defaults to pass (for example, true=true) then the autograder will automatically subtract 10 points from your score. You need to pass the tests you're supposed to pass and fail the ones you're supposed to fail in order to get full points.

- **Q:** My code doesn't default to pass but I'm still getting a 0/10?
  A: The most common reason for this type of failure is that your code is returning some value. You should have no return values or display any text.

- **Q:** I discovered a flaw in the autograder - what should I do?
  A: Please inform your TA and/or Brandon or Paola via email with the details. This is only the second semester the code was transitioned to python v.3 and we're using autograder. In the process last semester, we found a few flaws that were addressed. However, we know there could be more.