Homework 3 – Using HTTP

# Lab Information

## Due Date:

Homework 3 Dropbox Deadline

## Objectives/Goal:

To evaluate common security concerns using a custom high speed user-agent.

Hooli Inc, is a reputable web application service provider. This semester you will be serving as Hooli’s Web Application Tester.

By the end of this week we will complete our unit on HTTP and underlying web technologies. At this point you should be able to understand and automate a large portion of the activities you undertake on a daily basis. Your goal will be to use this knowledge to further your pentesting of Hooli. After you have develop your solutions you will be charged with scanning the Hooli infrastructure.

## Deliverables:

* An image or scan of the completed signoff sheet.
* Your writeup
* You code

## Table of Contents:

[Lab Information 1](#_Toc461713141)

[Activity 1: Extracting Data 2](#_Toc461713142)

[Activity 2: Crawling for emails 2](#_Toc461713143)

[Activity 3: Harvesting URLs 3](#_Toc461713144)

[Activity 4: More Intelligent Brute Forcing 3](#_Toc461713145)

# Activity 1: Extracting Data

HTML is a powerful markup language that is used in many different ways throughout the web. Because of this you often need some context as to what you are grabbing. Hooli wants to make sure you are up to the challenge of scanning their network so they want you to demonstrate you understand basic HTML processing

## Step 1: Teaching your agent – HTML Processing

Because of its versatility, processing HTML really requires a parser. Fortunately for you, the BeautifulSoup library, or your languages equivalent, allows just such a capability. BeautifulSoup allows you to use traverse the document object via various methods including XPATH. Demonstrate that you can use it by automatically extracting all the courses that have both numbers and names from the Computing Security program overview page (<https://www.rit.edu/programs/computing-security-bs>).

## Step 2: Teaching your agent - Threading

While manually going over one page is nice, often you’ll have to fetch resources external to your site. It will always be much quicker to parallelize this activity. Modify your user-agent to include threading so you can show Hooli that you can quickly save all the pictures from <http://www.rit.edu/gccis/computingsecurity/people> into a folder.

# Activity 2: Crawling for emails

While you were busy demonstrating your capability to quickly download photos, the security team realized that this technique can be used as part of their phishing campaign. They asked you to expand your tool to be able to scan large pages for email addresses.

## Step 1: Teaching your agent - Scope

Alright, alright your agent is getting pretty powerful, but it is still only limited to processing one page, websites consist of hundreds of pages. Modify your user-agent to be able to crawl an entire web site. To crawl a site you may wish to familiarize yourself with a few search algorithms such as IDDFS/BS/DFS. Use these search algorithms in conjunction with the links on each page to generate a tree to crawl. Your code should allow you to limit to a domain and a numerical depth of search. You may also wish to add the ability to scope based on URL path.

## Step 2: Teaching your agent – SSL/TLS

You are going to need to interact with Hooli’s ultra-secure website. In order to do that you’ll need SSL/TLS to be supported. Implement SSL/TLS support within your user-agent.

## Step 3: Crawling for emails

Now that your user-agent can use BeautifulSoup, is threaded, supports TLS, and you can limit by scope – you should be set to crawl for email addresses. Use your user-agent to scan https://www.rit.edu to a depth of 4, for emails.

# Activity 3: Harvesting URLs

Hooli has challenged you with doing some recon their site. You know like many sites they often have goodies hidden in places that are not hyperlinked from the main site. You want to create a scanner for these, but before you can you need to create a list of where things might be hiding. Use your new spidering capabilities to generate such a list.

## Step 1: Teaching your agent – Multiple Targets

You’ve done well, now you need to do more than just one domain. In this case we’re not looking for email addresses but actual paths. While there are existing dictionaries of common directories, for our next step we’re going to need to build our own. Use the same concept as your previous activity to visit at least 25 sites on the companies list provided (companies.csv). Visit them to a depth of at least 4 and record their URLs

## Step 2: Extracting Data

Now that you have thousands of URLs you should extract the data you need. You will be tasked with finding hidden directories and pages. Parse your URLs into an output file such that it can be used as input for the next stage.

# Activity 4: More Intelligent Brute Forcing

Hooli understands security, and they want only their engineers with their applications to be able to access the next flag. To that end they’ve put a little challenge in place that only their application can solve. Can you get by it and get to the third flag?

## Step 1: Scanning for gold

Using your new list of possible paths and pages, scan the new Hooli website http://52.23.205.104 to see if you find any folders or files that were previously not available from the homepage. You may wish to also spider the page such that you can brute force known existing folders. Note down any of these files that might represent security issues in your writeup.

## Step 2: Writeup

* Is there certain information about the webserver that you can discern based on what files you can access?
* Are there any ways to improve the speed of your scanner?
* How can response codes be used in order to more efficiently search your site?
* Are there any common naming patterns that you might expect would yield positive results?

Signoffs

## Activity 1.1 – Processing HTML

Show your instructor that you are able to create a directory of classes and class numbers using an HTML parsing suite

## Activity 1.2 – Speeding up requests

Show your instructor that you are able to download all the pictures on the CSEC faculty page simultaneously using threads.

## Activity 2 – Crawling for emails

Show your instructor that you are able to crawl [https://www.rit.edu](https://www.rit.edu/) (and only RIT.edu) for email addresses. Also show that you are able to stop at a given depth.

## Activity 3 – Harvesting URLs

Show your instructor that you are able to crawl multiple websites and collect their URLs path’s and pages into a list.

## Activity 4 – Brute Forcing

Demonstrate that you can scan undertake a dictionary brute force scan of Hooli’s page.