

Design Manual

Student: Sean O'Connor – C00224424

Supervisor: Richard Butler

Cybercrime & I.T. Security – CW_KCCYB_B

Institute of Technology Carlow

Contents

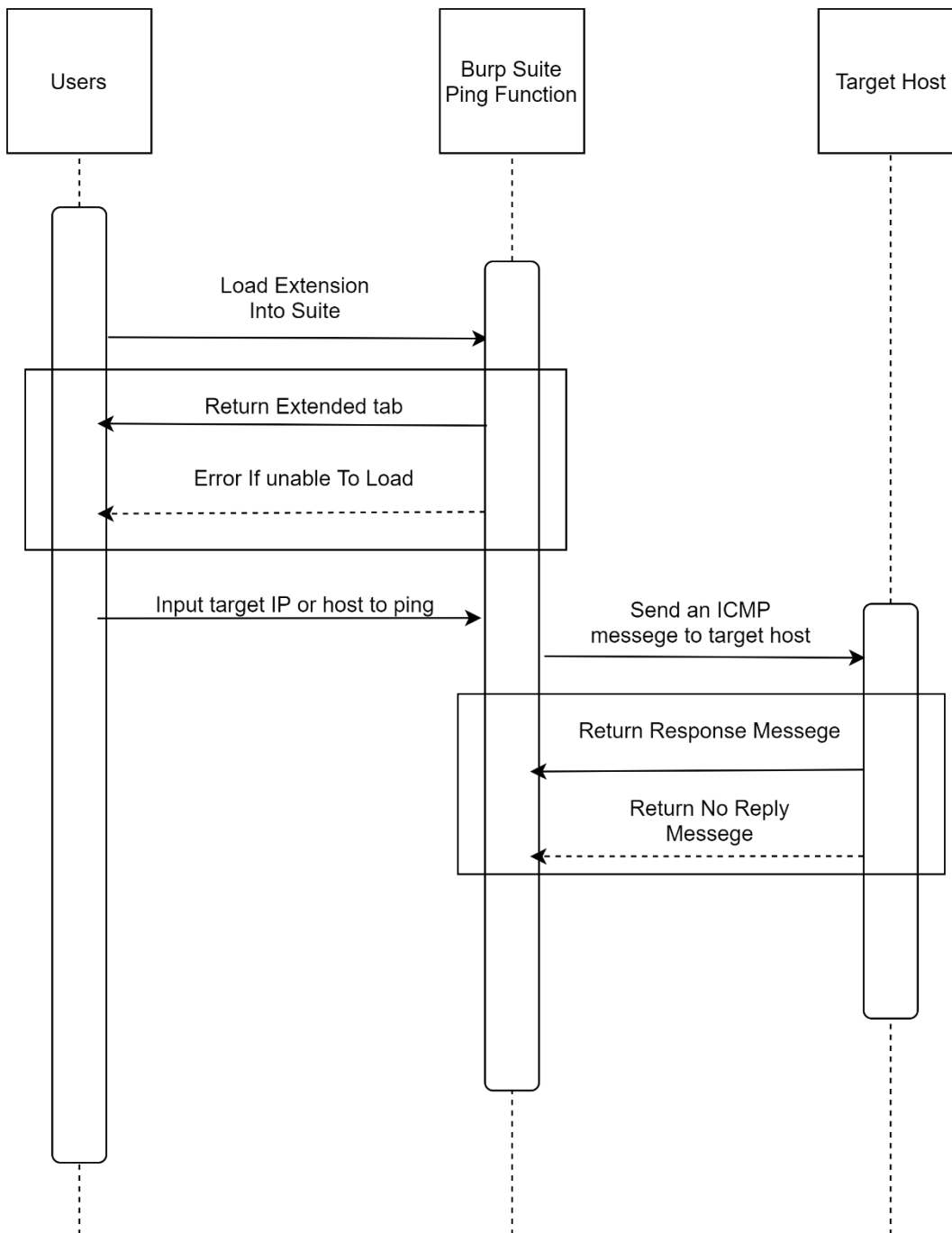
Introduction	3
Sequence Diagrams	3
Ping Function	3
Ping Sweep Function	4
Threaded Port scan.....	5
Subdomain Crawler	6
HTML Scraper	7
User Interface Examples	8
GUI Code.....	8
Ping and Ping Sweep	9
Threaded Port Scan.....	10
Subdomain Crawler	11
HTML Scraper.....	12

Introduction

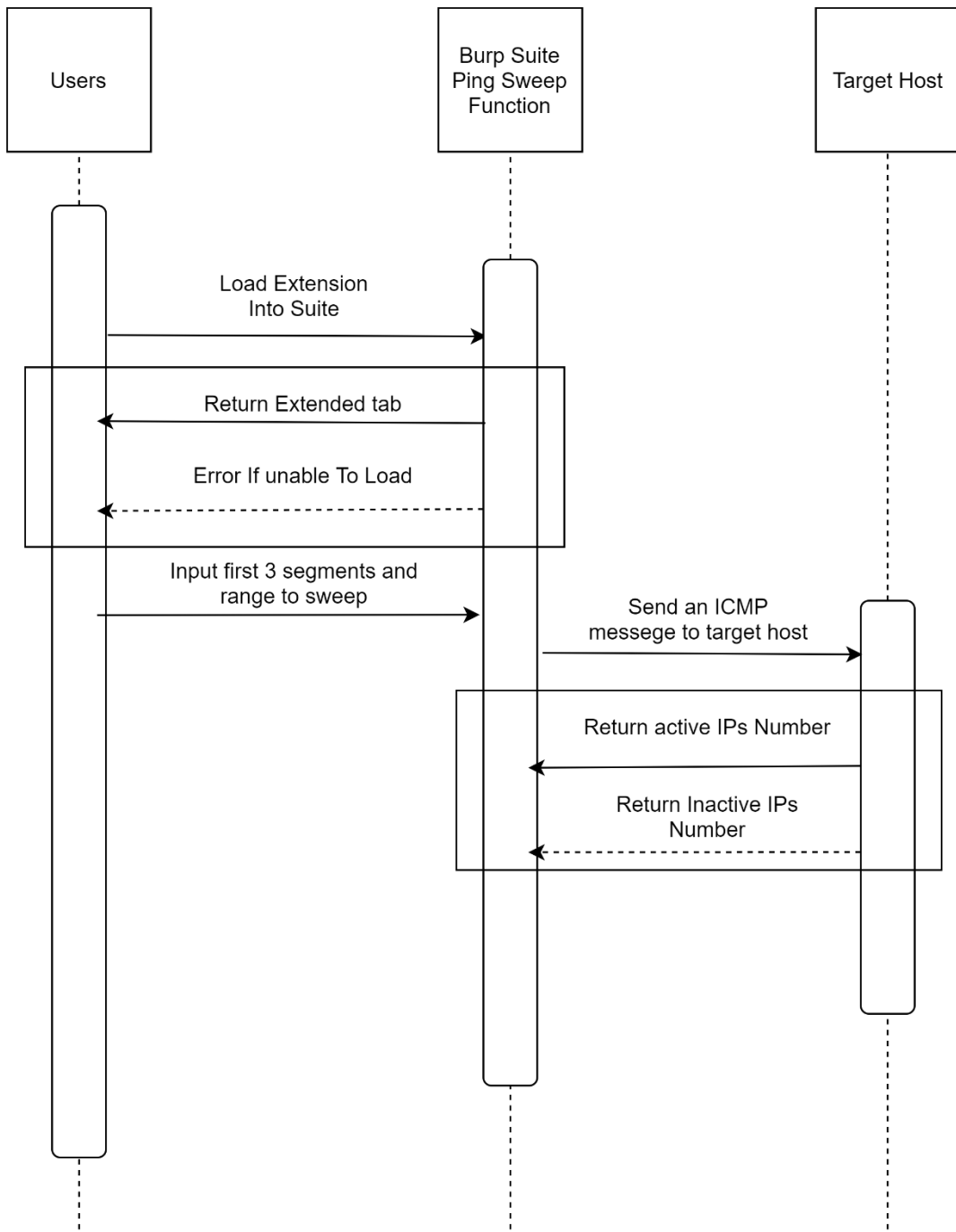
In this document we will be going through the design specification for the function tabs for the extensions we will be creating. We will go through each of the functions and their sequences. After this we will then go through a brief description of the layouts of the function tabs along with a simple wireframe mock-up of the layout.

Sequence Diagrams

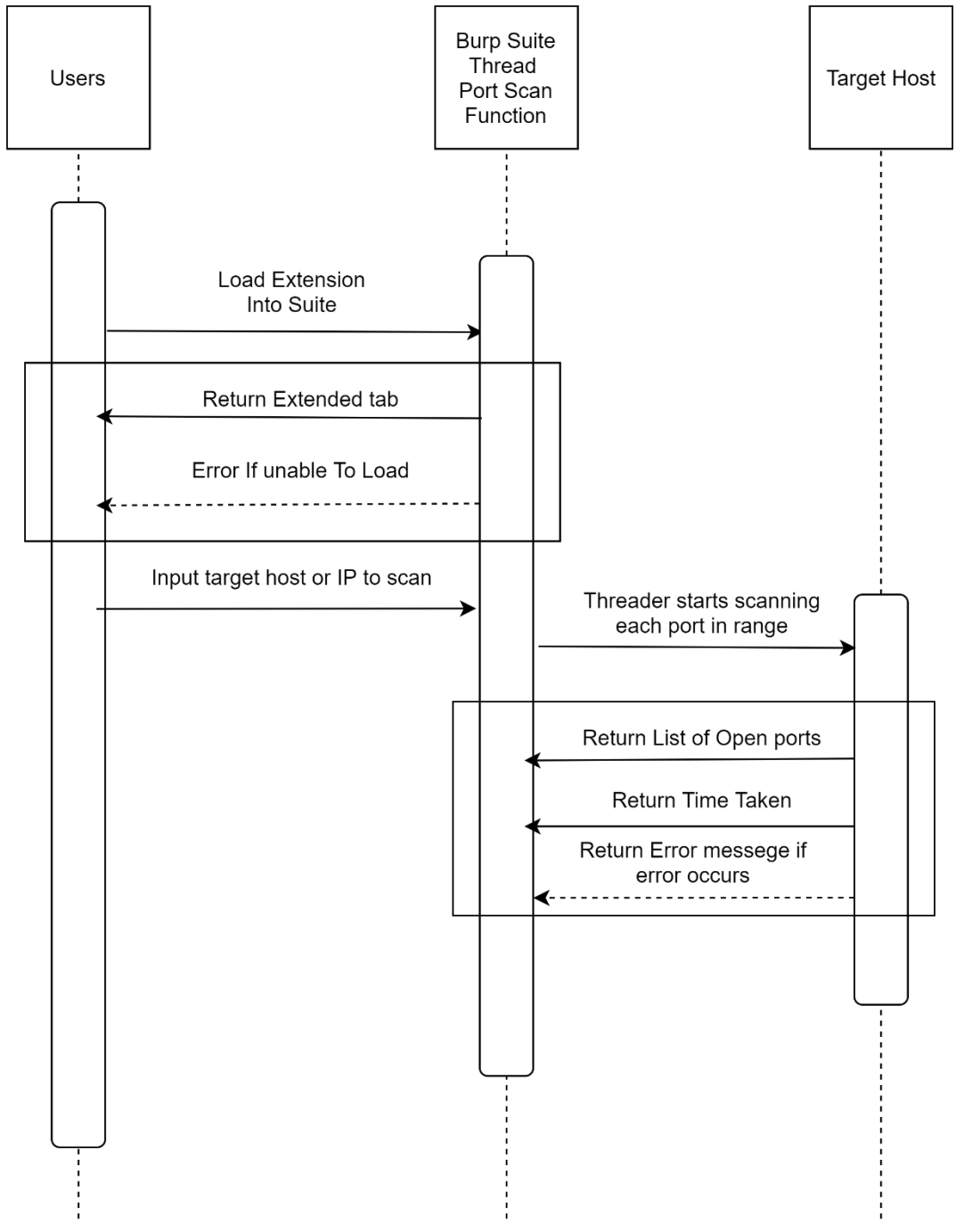
Ping Function



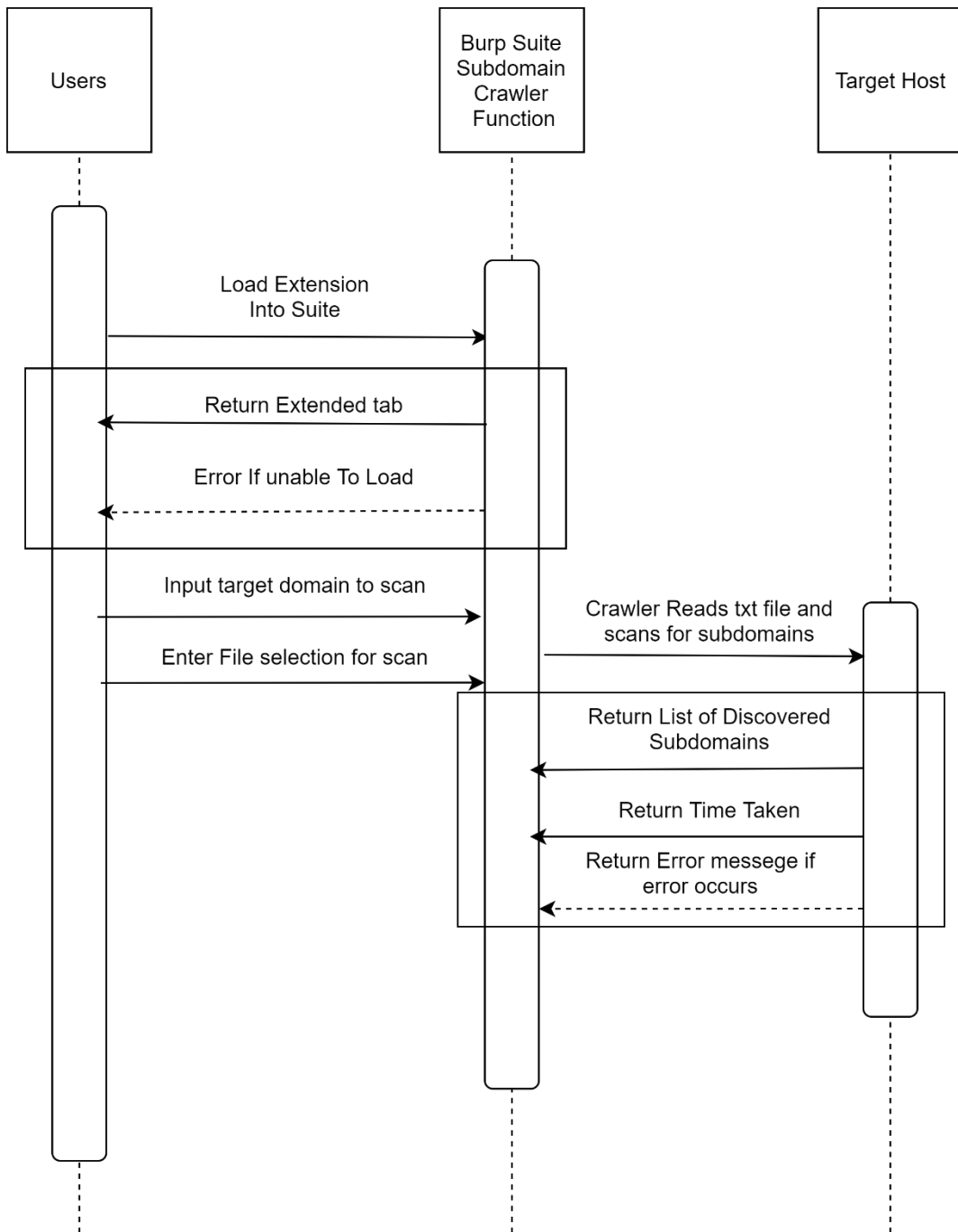
Ping Sweep Function



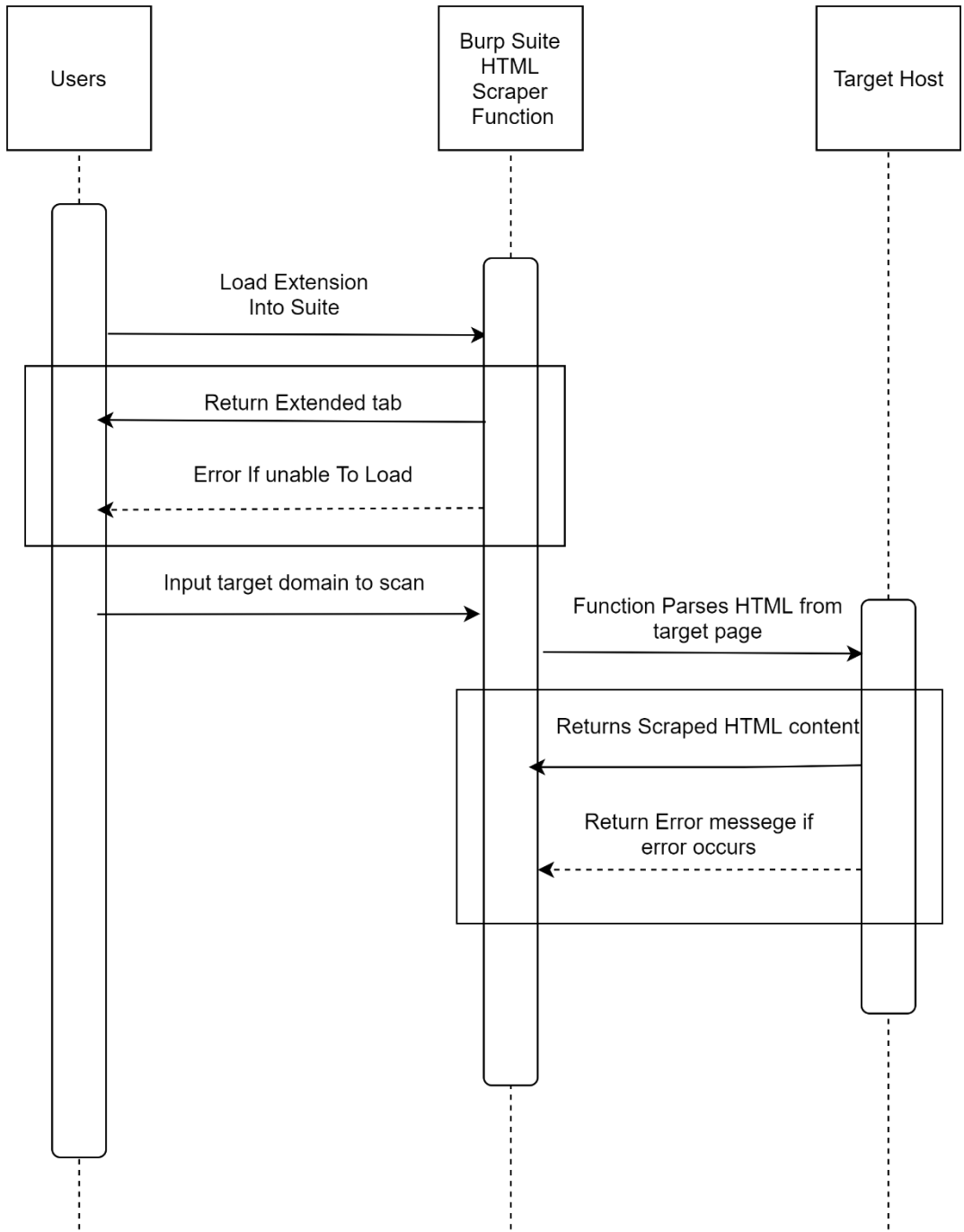
Threaded Port scan



Subdomain Crawler



HTML Scraper



User Interface Examples

GUI Code

For the actual GUI creation, the code was the same the whole way through. It firstly consisted of creating the actual tabs and sub tabs for the input and result boxes to sit on. This was done using a mix of Burps callbacks feature and using java swing to create JPanels and JTabbedPanels.

(See Fig. 1)

For the input and output boxes, I used JPanels and Boxes to create the necessary input and output boxes. For all the input and output boxes the code repeated itself except for box sizes and labels.

(See Fig. 2)

The Final step in the GUI creation was to make the buttons, using JPanels and JButtons to create and link these buttons to the function. After this the buttons were added to the appropriate tab.

(See Fig. 3)

```
class BurpExtender(IBurpExtender, ITab):
    def registerExtenderCallbacks(self, callbacks):
        sys.stdout = callbacks.getStdout()

        ### MAIN TAB ###
        self.callbacks = callbacks
        self.callbacks.setExtensionName("Project Tab")
        self.tab = swing.JPanel(GridLayout())

        ### SUB TABS
        tabbedPane = swing.JTabbedPane()
        self.tab.add("Center", tabbedPane)

        tab1 = swing.JPanel()
        tab1.layout = GridLayout(2,3) #Gives tabs a grid layout
        tabbedPane.addTab("Ping", tab1)

        tab3 = swing.JPanel()
        tab3.layout = GridLayout(2,1)
        tabbedPane.addTab("Threaded Port Scanner", tab3)

        tab4 = swing.JPanel()
        tab4.layout = GridLayout(2,3)
        tabbedPane.addTab("Sub Domain", tab4)

        tab2 = swing.JPanel()
        tab2.layout = GridLayout(2,1)
        tabbedPane.addTab("Scrapper", tab2)
```

Fig. 1

```
### INPUT BOX ###
inputPanel = swing.JPanel()
boxVert = swing.Box.createVerticalBox()
boxHori = swing.Box.createHorizontalBox()
inputLabel = swing.JLabel("Target host/IP")
boxHori.add(inputLabel)
boxVert.add(boxHori)
boxHori = swing.Box.createHorizontalBox()
self.inputArea = swing.JTextArea('', 1, 25)
self.inputArea.setLineWrap(True)
boxHori.add(self.inputArea)
boxVert.add(boxHori)
inputPanel.add(boxVert)
```

Fig. 2

```
### BUTTONS ###
## button creation
buttonPanel2 = swing.JPanel()
buttonPanel2.add(swing.JButton('Scrape', actionPerformed=self.scrapper))
tab2.add(buttonPanel2)

buttonPanel3 = swing.JPanel()
buttonPanel3.add(swing.JButton('Scan', actionPerformed=self.scan))
tab3.add(buttonPanel3)

buttonPanel4 = swing.JPanel()
buttonPanel4.add(swing.JButton('Lookup', actionPerformed=self.lookup))
tab4.add(buttonPanel4)
```

Fig. 3

Ping and Ping Sweep

Below is a screenshot of the Ping and ping sweep functions. Instead of having a separate tab for each function, it made more sense to have them on the same tab.

The tab consists of two rows, the input row being on top, and the button/results row on the bottom. Users can enter in a target IP on the leftmost input field, press the “Ping” Button directly below and get their results in the text box to the right of the button.

For the sweep, there is an input field, for the first three segments of the target IP, then two smaller fields for the start range and end range. Below this is another button marked “Sweep” and a results box.

The screenshot shows a web application interface with a navigation bar at the top containing four tabs: "Ping", "Threaded Port Scanner", "Sub Domain", and "Scrapper". The "Ping" tab is currently selected. Below the navigation bar, there are two rows of input fields and buttons. The first row contains four input fields: "Target host/IP" with the value "test", "Target IP (First 3 segments only)" with the value "1.2.3", "Range Start" with the value "1", and "Range End" with the value "4". Below these input fields, there are two buttons: "Ping" and "Sweep". To the right of the "Ping" button is a text box labeled "Result" containing the text "test", "test", "test", and "test". To the right of the "Sweep" button is a text box labeled "Sweep Result" containing the text "test" and "test".

Threaded Port Scan

Below is a screenshot of the Threaded Port Scanner tab. The tab consists of an input text box, the button used to execute the script, the results box and another text box for the time taken for the file to execute.

The screenshot shows a web application interface with a tabbed menu at the top containing 'Ping', 'Threaded Port Scanner', 'Sub Domain', and 'Scraper'. The 'Threaded Port Scanner' tab is active. The interface includes a 'Target host/IP' input field with the text 'test' and a 'Scan' button. Below this, there are two output boxes: 'Open Result' with the text 'test' and 'Time Taken' with the text '1234'.

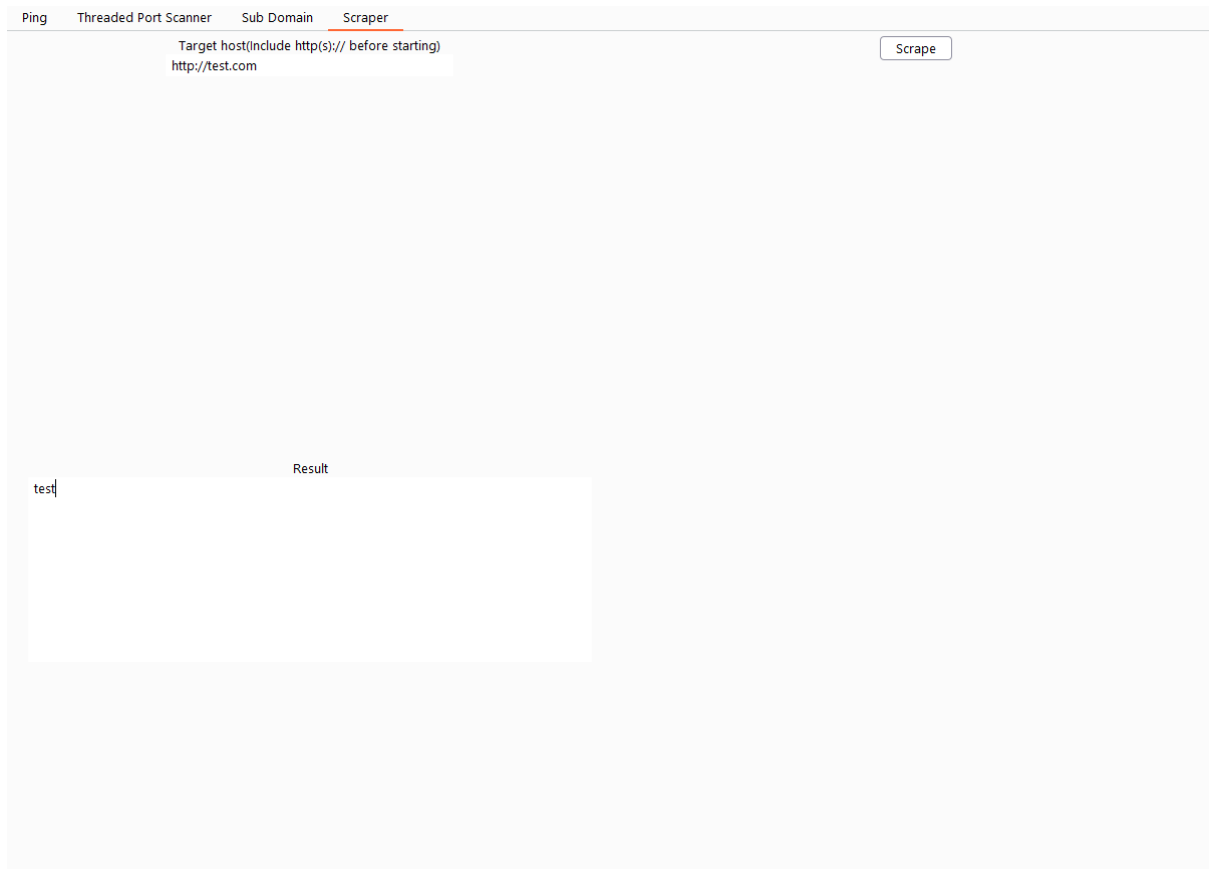
Subdomain Crawler

Below is a screenshot of the Subdomain Crawler function tab. The tab consists of a input field for the target domain, another input field on the top right for file selection, a button in the middle for execution of code, a result box on the bottom left, another box in the middle for the time taken, and finally a button on the left for clearing the results box.

The screenshot shows a web application interface with a navigation bar at the top containing four tabs: "Ping", "Threaded Port Scanner", "Sub Domain" (which is highlighted with a red underline), and "Scraper". Below the navigation bar, the interface is divided into several sections. On the left, there is a "Target host/IP" input field containing the text "test.com". To its right is a "Lookup" button. Further to the right is a "Select file 1,2 or 3" input field containing the number "1". Below these input fields, there is a large, empty rectangular area. At the bottom of the interface, there are three main components: a "Result" box on the left, a "Time Taken" box in the middle containing the text "1234", and a "Clear" button on the right.

HTML Scraper

Below is a screenshot for the HTML Scraper function tab. The tab consists of a input box for the target to be scraped, with instructions, the button which will execute the code and finally the result box where our output will be displayed.



The screenshot shows a web application interface with a navigation bar at the top containing four tabs: "Ping", "Threaded Port Scanner", "Sub Domain", and "Scraper". The "Scraper" tab is currently selected and highlighted with a red underline. Below the navigation bar, there is a main content area. At the top of this area, there is a text input field with the placeholder text "Target host(include http(s):// before starting)" and the value "http://test.com". To the right of the input field is a button labeled "Scrape". Below the input field, there is a section labeled "Result" which contains a large, empty text area. The word "test" is visible in the top left corner of this result area, likely representing the start of the output or a label for the content.