EE / CprE / SE 491 – sdmay 21-17
Webserver Attack Blocking AI
Report 3

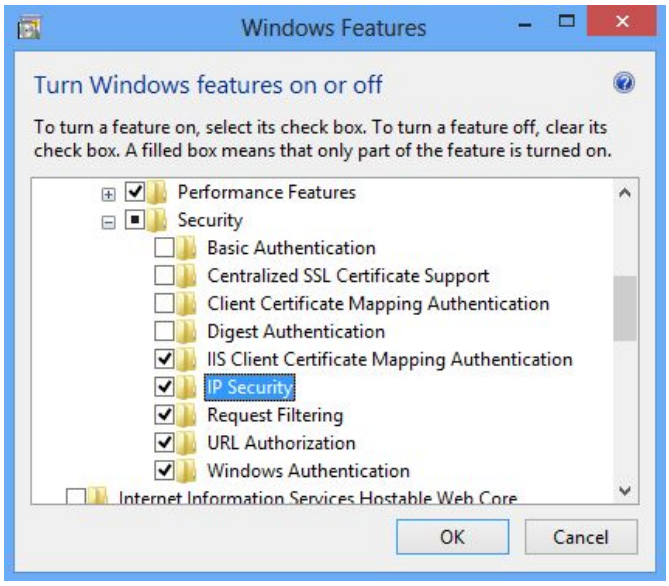
02/22/2021 – 03/01/2021
Client: Andrew Dakin (Cylosoft)
Faculty Advisor: Douglas W. Jacobson

Team Members:

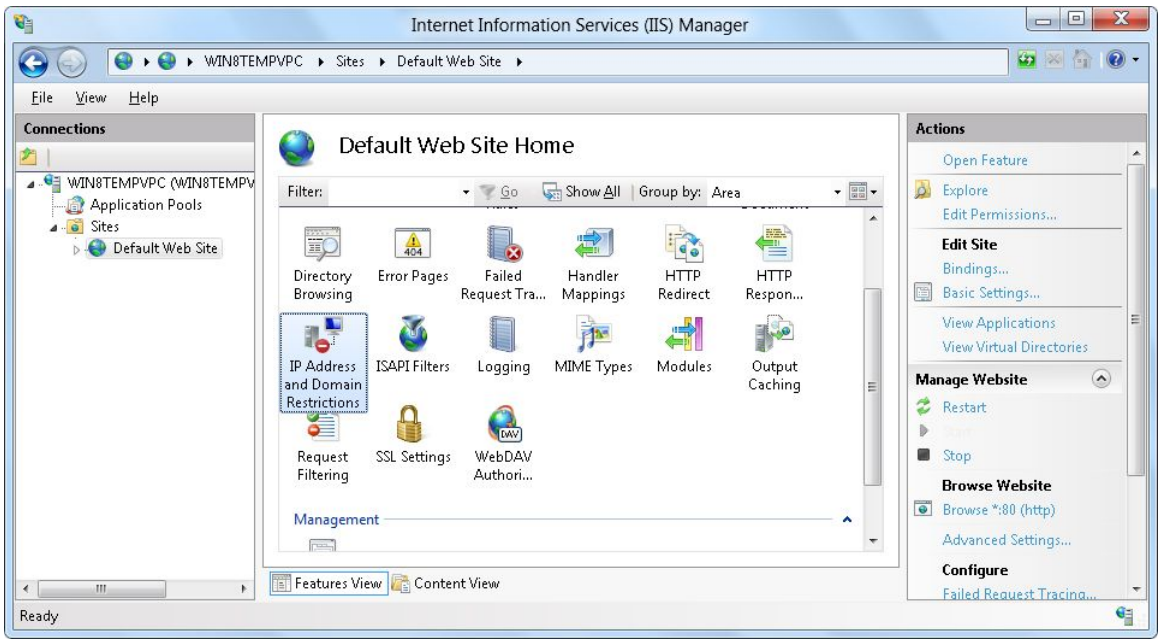
Megan Hill ---------- Website Engineer
Jamie Sampson ---------- Security Engineer
Emily Young ---------- Database Engineer
Andrew Marek ---------- Administrator / Software Engineer
Paul Degnan ---------- Automation Engineer

Past Week Accomplishments

- IP Security - Jamie
    - Integrated the NLog package into the project to obtain a more robust logging system than printing to the console.
        - Created a separate class that builds and tears down the logger
        - Created a readme on how to use the package
    - Connected with IIS
        - Found out that to make the calls that were necessary to change security, it was necessary to run IDE as admin.
        - Identified the location in IIS where the IP rules were being stored per website.
        - Connected locally / to the default website
    - Tested functionality to add, remove, and clear IP addresses
        - Shifted IP Security class from singleton to normal class

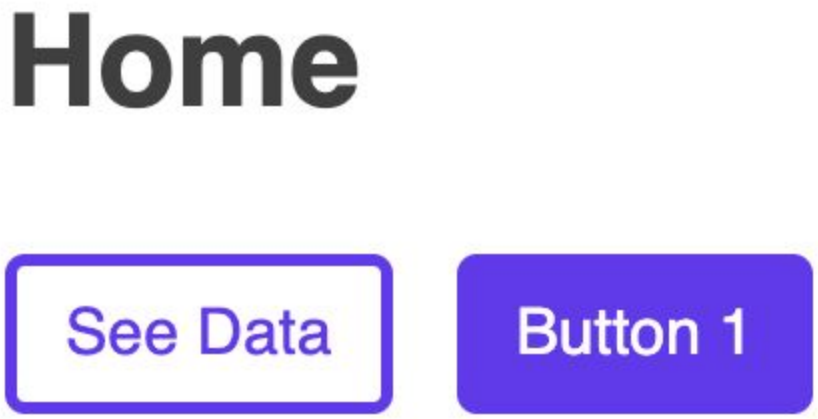


Windows feature selection menu to get IP Security module.



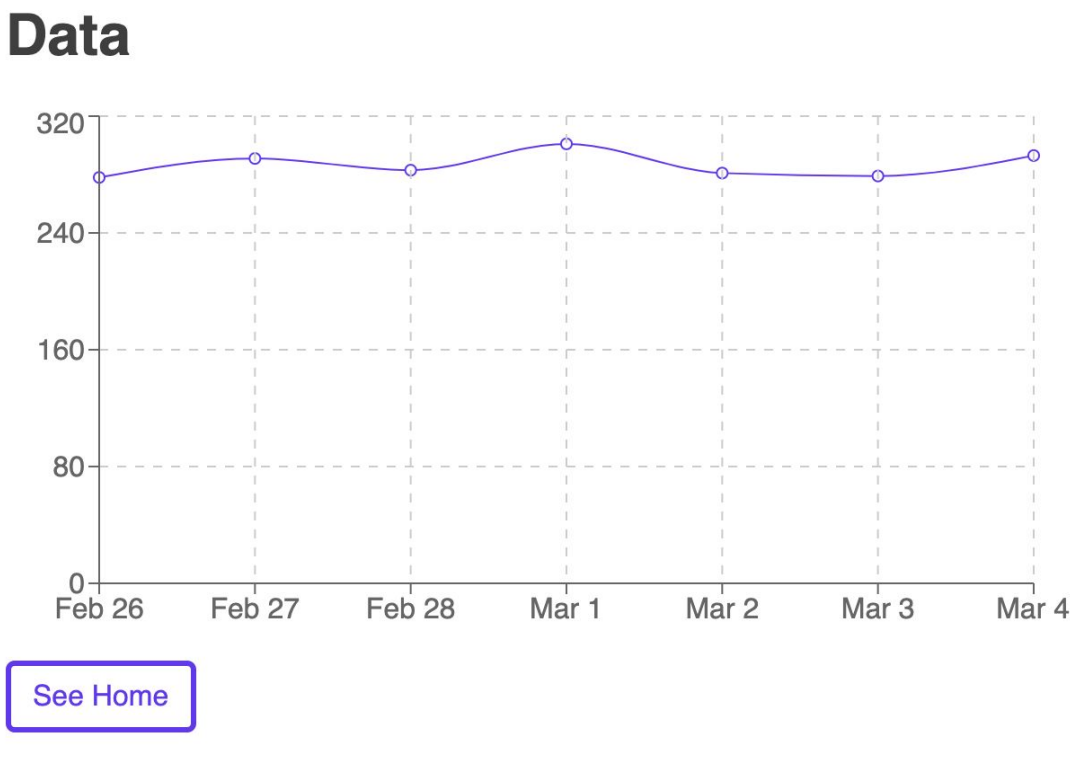
IIS UI dashboard for default website identifying location of where the IP addresses that are blocked are stored.

- Website - Megan



  - Home Page
    - Basic components (2 buttons)
    - Global styling (purple and font style/color)
  - Adding Testing to the project for 100% code coverage
    - Backend testing using jest as a testing library
    - Front end testing using jest and react shallow renderer
  - Added Linting to the project to enforce code cleanliness standards

        - Using eslint for most linting
        - Using npm package get-off-my-lawn for a basic set of lint rules
        - Fixed existing code to follow linting rules
    - Displaying Data on website
        - Use rechart to display data from charts
        - While data comes from the backend it is mock data for now, until we connect to the database

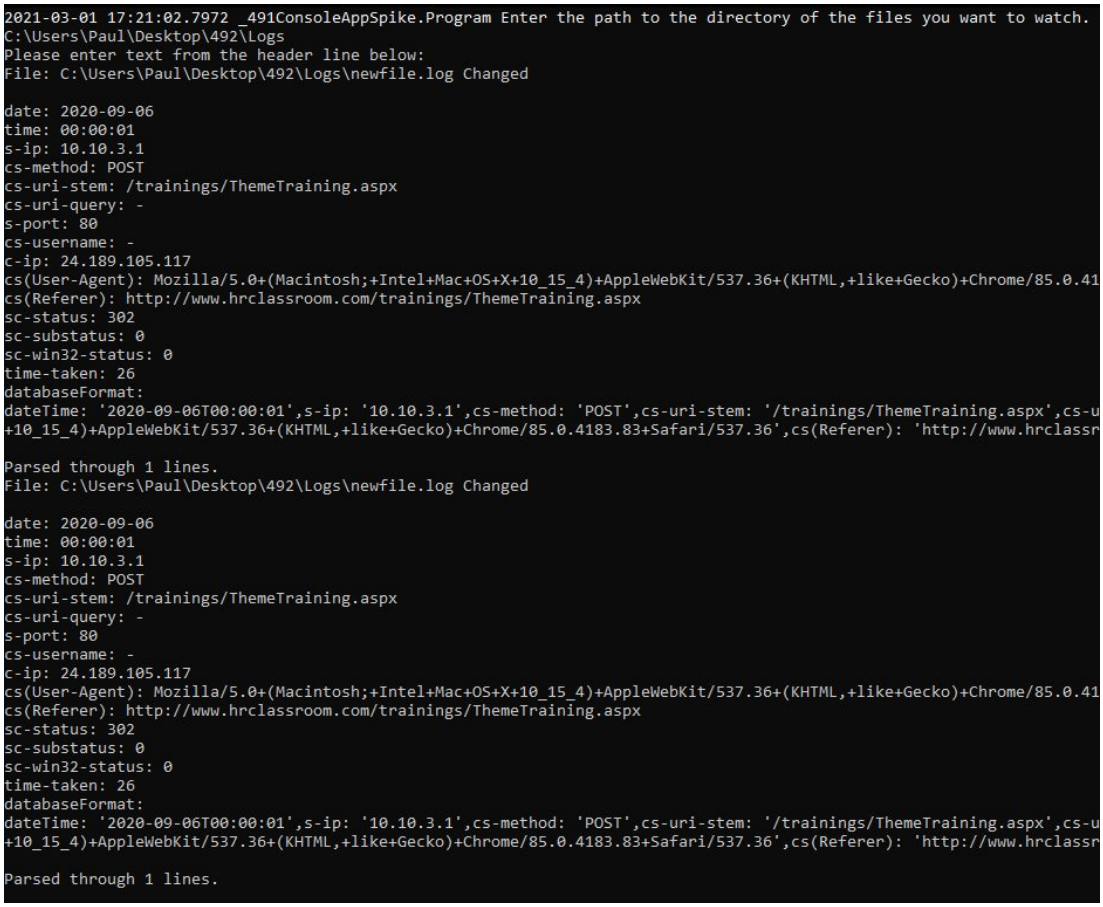


- General refactor / general changes - Andrew
    - Change IP we capture to cf-connecting-ip as per client's information.
    - Parse IIS headers dynamically such that a log's header could be different and the program still works.
    - Create a PR template.
    - Change responsibilities of classes to follow single responsibility principle (somewhat).
    - Cleaned up variable naming, trying to follow C# conventions.

- File Watcher - Paul
    - Fixed a bug where the file watcher would error out after the first read of a file

        - Also took out the temporary ReadLine that let testing continue for other parts of the file watcher
    - Continued work on making the Log Parser only run on new lines of the changed file
        - I believe the way to go about this will be adding in an OnAddition method that runs every time a file is added to the watched directory. The reason I believe this is necessary is because the Log Parser needs to know whether or not it is working on the same file that has been changed multiple times, or if it is an entirely new file. This will only work if the assumption that there is one log file per day is correct. Will need to get clarification and a better understanding of the file system especially if there are subdirectories that need to be watched too.

Picture of Progress: The File Watcher events run every time there is a change



- Database - Emily
    - Located a reliable testing environment
        - As mentioned in the previous report, I was having a range of difficulty connecting to the database that I had set up with Azure. I was beginning to fall behind on my code output as a result

        - This week I was able to locate a strictly web-based environment that allows me just paste my code and test the queries
    - Update Insert and Delete methods
        - The Insert and Delete methods are now more robust and up to date with current changes to our main program's parser
        - Testing is still in progress. There are a couple of items that are still being inserted incorrectly into the database
    - Create Get method
        - I have the skeleton code for a get method that we are hoping to implement in the website part of our finished product
        - The plan is to have it pull the IP by date

Pending Issues

- Combining each part of our code into the master branch. - Everyone
- Thing 2

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Megan Hill	Adding Testing, Linting, and other features to existing website infrastructure	5	14
Jamie Sampson	Added NLog package for logging. Connected with IIS and identified where settings were. Tested add, remove, and clear IP addresses.	8	15
Emily Young	Updated Insert/Delete. Created the skeleton of the Get method. Located and utilized a reliable testing environment.	6	13.5
Andrew Marek	Finishing up general refactor -- attempting to follow C# conventions and good OOP practices.	6	12
Paul Degnan	Fixed bug where file watcher tried to run log parser on header of IIS logs. Fixed bug where file watcher only ran for a single change in a file.	3	10

Plans for Coming Week

- IIS/IP Security - Jamie
    - Merge code into master
    - Connect IIS/IP Security with FileWatcher using found path to log files
    - Connect IP Security w/ database code

- Refactor code architecture (rename items, add/remove folders) - Jamie
- Connect Website to database - Megan
- File Watcher - Paul
    - Make sure the same IIS log lines are not being read over and over.
    - Automate the process of a new log file that needs to be watched being created.
        - In regards to this process, figure out if it is for sure necessary. I believe it will be to make sure the same IIS log lines are not being read over and over.
    - Look into making the file watcher watch subdirectories.
- Fix Insert to work 100% of the time - Emily
- Finalize Get method - Emily
- Create YAML config file for program such that one can choose frequency of logs, add new features on demand, etc.- Andrew