Defensive Strategies Against Low & Slow Attacks in HTTP2

Project Topic, Initial Decisions & ResearchPhase

Topic

Low & Slow Attacks in HTTP/2

Objective

To decide between developing a defensive or offensive strategy against Low & Slow attacks in the HTTP/2 protocol.

Literature Review

Conducted an extensive literature review using various sources, including Google Scholar and ChatGPT.

Key Study Identified

Focuses on anomaly detection by identifying delays between frames in Low & Slow attacks on HTTP/2.

Research Outcome

Introduced new methods for detecting prolonged and subtle network attacks, leading to the choice of a defensive solution.

In-Depth Exploration of HTTP

Exploring HTTP and Focusing on Attacks and Defenses:

- **Comprehensive Study of HTTP**

  Undertook a comprehensive study of HTTP, examining all its versions, encryption methods, and data traffic protocols.

- **Advantages Analysis**

  Analyzed the advantages of HTTP and its various versions to understand its capabilities and limitations.

- **Focusing on Attacks and Defenses**

  Delved deeper into potential attacks and defense mechanisms within HTTP environments.

- **Researched Attack Types**

  Researched the types of attacks that could occur and the strategies for addressing them, focusing on the vulnerabilities and potential uses in attack scenarios.

1

In-depth Reading of the Paper

Detailed review of the selected paper to fully grasp the research specifics.

2

Focused on understanding the insights and methods proposed for anomaly detection in attacks.

3

Communication with the Paper's Author

Sent multiple emails to the article's author to obtain the original code and auxiliary functions.

Beginning of Code Development

Transition from Theory to Practice

Started the coding process based on the pseudocode presented in the paper.

Developed key functions as described in the paper: a function for data processing and analysis, and a function for anomaly detection and classification.

Physical Implementation of Code

Wrote the actual code while adhering closely to the pseudocode.

Physically implemented processes that were theoretically described in the paper.

Code Implementation

and Anomaly Detection

1. **Focus on Delay Detection**

   Started by focusing on detecting delays to first ensure the code's ability to identify such anomalies.

2. Developed the code to handle delay detection before addressing more complex issues like mismatch.

3. **Data Acquisition for Validation**

   Finding a suitable dataset for testing the code proved to be challenging as PCAP recordings and data on HTTP/2 are scarce.

4. Searched various sites, attempted to record data ourselves, and contacted researchers globally.

5. Eventually found data that was sufficiently good but not perfect.

Problem Solving and Code Improvement

Addressing Pseudocode Issues:

- Developed a function that splits HTTP/2 streams directly from the PCAP and automatically
- Developed a function that splits HTTP/2 streams directly from the PCAP and automatically classifies them.

Enhancing Data Processing and Analysis:

- Chose not to use the cumbersome method proposed by the paper's author for handling streams efficiently.
- Created a new array at the beginning of each function and updated external files at the end to
- maintain data integrity.

Improving Storage and Management of Streams:

Developed a function to save all streams, renaming them and numbering them in chronological

Enhancements in Detection

and Code Work

1. **Successful Delay Detection**

   Successfully identified high amounts of delays during attacks, which enabled further improvements in anomaly detection.

2. **Handling MISSMATCH and Improving Anomaly Detection**

   Advanced to detecting and addressing MISSMATCH in the best possible way.

3. Added and refined code to achieve accurate and swift anomaly detection.

4. **Testing and Code Evaluation**

   Run the code on attacks used by the paper's author and on additional attacks we found.

5. Despite not finding a perfect dataset, which resulted in some deficiencies, the detection level was

6. The code is well-written, efficient, and well-organized, yielding very good results.

Writing the Paper and Creating a Git Repository

Paper Writing

Authored a detailed paper about our research and findings on detecting Low & Slow attacks in HTTP/2.

Git Repository Creation

Established a well-organized and documented Git repository containing all the code files and associated documentation.

Included a relevant README file that explains the purpose, the methods used, and how to utilize the developed code.

Description of a primary heading

Challenges Encountered During the Project

Learning from Existing Research and Code

Difficulty in understanding a research paper and code not written by us, required extensive time to grasp functions and logic.

Slow communication with the paper's author required patience and waiting for responses.

Time and Psychological Challenges

Working under pressure due to academic load and psychological stress, as well as reserve duty during wartime, which hindered focus on the project.

Finding a Dataset for HTTP/2

- The main challenge was finding a suitable dataset with PCAP recordings for HTTP/2, which was scarce and took a lot of time and resources.
- Dealing with disappointments and deficiencies in the information obtained.

Handling PCAP Data:

- Technical difficulty in splitting and processing the PCAP to identify specific streams and flows.
- It took time to find the right method for efficiently working with the data.

Conclusions and Summary of the Project

1 Overall Performance

We believe we have done well despite the challenges and imperfections.

2 Project Experience

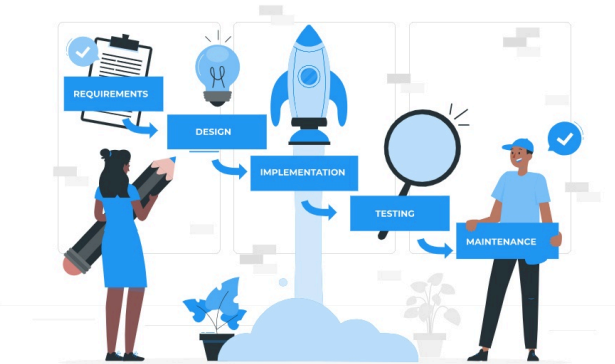
The project was interesting and educational, deepening our understanding in research, coding, and communication with researchers.

3 Key Learnings

Developed significant knowledge in HTTP traffic, particularly in HTTP/2.

4 Research Gaps

Recognized the deficiencies in the existing data on HTTP/2 and the need for further research.



5 Acknowledgments and Thanks

Special thanks to Amit Dvir for his guidance and support throughout the project.

6 Credit

Credit to the paper's author, Nikhil Tripathi, whose work was a starting point for our research.

7 Final Thanks

Thank you very much, Naor Ladani, Itamar Cohen, and Tovia Smadar.