The 21th Winona Computer Science Undergraduate Research Symposium

April 19, 2021

9:30am to 12:00pm

Join Zoom Meeting
https://minnstate.zoom.us/j/94259024563

Winona State University
Winona, MN

Sponsored by the Department of Computer Science

at Winona State University

WINONA STATE UNIVERSITY
Computer Science Department
http://cs.winona.edu

Table of Contents

Testing Relative Performance of MERN, MEAN, and LAMP Web Development Stacks

Randall (RJ) Bradach
175 W Mark St, Winona, MN 55987
1-952-923-3515
rjbradachcs@gmail.com

ABSTRACT
The goal of the research project was to answer the question, "Which web development stack is most efficient and powerful?". This is an important question to answer as it affects every website created today. Finding the best web development stack is incredibly important as it can pre-determine how successful the future of the website you develop will be. The scope of this article pertains to web development as a whole, and specific combinations of different web development technologies. The combinations of web development technologies are known as "stacks". The three being compared include the MERN (MongoDB, Express, React, NodeJS), MEAN (MongoDB, Express, Angular, NodeJS), and LAMP (Linux, Apache, MySQL, PHP) stacks. The method of analysis revolves around testing the time complexities of the stacks with multiple large database queries, using computationally intensive algorithms. After running database queries involving differently sized quantities of data, and measuring the time intervals for each stack's performances from those queries, experimental results were compared against each stack.

General Terms
Algorithms, Management, Measurement, Documentation, Performance, Design, Experimentation.

Keywords
Stack, Development, Database.

1. INTRODUCTION

With respect to web development, it is of utmost importance to determine which software you include while building your projects. Without searching for the right software could lead to wastes of time, resources, and money. The conquest of searching for the best combinations of software has evolved into denoting such combos as "stacks". These stacks are composed of two distinct parts. The first part included software that falls under the "front-end" web development umbrella. The front-end is commonly referred to as "the looks" of a website. This umbrella encompasses markup and web languages such as HTML, CSS, and Javascript; and single page applications made from frameworks like React, Vue.js, and AngularJS. These software among many other fragments and subgenres of web development make up the front-end half of a technology stack. The other half of the stack is the counterpart to the front-end, being the back-end, otherwise known as "the logic". The back-end umbrella encompassing scripting languages (PHP, Python, Node.js) and compiled languages (Java, Go, C#).

Next we preview the technology stacks used for analysis in this project. These stacks all prominently feature acronyms that can be split into each distinct and irremovable software. All software included in the stacks serve a useful and distinct function.

1.1 Web Development Stacks
Here we define the three web development stacks that help compose the experiment.

1.1.1 MERN Stack

The MERN stack is composed of the following technologies.

MongoDB – a document database with the scalability and flexibility that you want with the querying and indexing that you need [13]

Express.js - Fast, unopinionated, minimalist web framework for Node.js [9]

React - A JavaScript library for building user interfaces [10]

Node.js - an asynchronous event-driven JavaScript runtime

1.1.2 MEAN Stack

The MEAN stack is composed of the following technologies.

MongoDB - a document database with the scalability and flexibility that you want with the querying and indexing that you need [13]

Express.js - Fast, unopinionated, minimalist web framework for Node.js [8,9]

Angular- is a JavaScript-based open-source front-end web framework mainly maintained by Google [1]

Node.js - an asynchronous event-driven JavaScript runtime [8,9]

Proceedings of the 16th Winona Computer Science Undergraduate Research Seminar, April 19, 2021, Winona, MN, US.

1.1.3 LAMP Stack

The LAMP stack is composed of the following technologies.

Linux - A family of open-source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on September 17, 1991, by Linus Torvalds [1]

Apache - The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards [4]

MySQL - A relational database management system

PHP - A general-purpose scripting language

Unlike the other two stacks which are built from client-side and server-side software. The LAMP stack left a crucial point of comparison between this stack and the other two. There must be a front-end counterpart on this stack. For this case, I used plain, natively understood languages HTML, and CSS to make up the missing pieces.

1.2 Hypothesis

The MERN web development stack is more performant timewise, relative to the MEAN and LAMP stacks.

2. BACKGROUND RESEARCH

We did not find a significant amount of peer reviewed literature that compared these web development stacks and their performance online. However, this proves that this article will serve the unique purpose of determining which web development stacks do best at which database scale.

Despite the lack of peer reviewed research for this topic, there are articles onlines that speak on these stacks and how they compare.. As a start to this research project, I will review a couple findings. From the article “Best Stacks For Web Development” written by systango, it is shown that most people would choose React over Angular. This is based on a survey, rather than empirical evidence to support that React performs better, but it is worth noting that people prefer using React within their development, which could imply that the MERN stack is more performant than the MEAN stack. Besides that, it is shown that plain old vanilla javascript is almost twice as fast as React and Angular, which might prove to make the LAMP stack the winner out of the three. However, this timing discrepancy is only based on frontend performance [3]. Also, despite favoritism of developers, there are many greater factors for why you should choose specific technology for a project. When making decisions on which stack to use, don’t pick one metric, include a multitude of factors beyond just speed or testimony. Make informed decisions.

3. METHODOLOGY

This project is an experimental research-based project. The aim is to conduct testing methods to collect data and to analyze and compare the three stacks introduced. The methodology to follow this idea will predicate on creating three programs, one for each stack, that perform functionality as similar as possible. These programs will produce results from each stack that significantly prove one stack is greater than the other two in performance.. The broad goal is to extract meaningful metrics and subsequently the overall differences between the stacks, rather than comparing each individual stack element to each other, between stacks. Since the composed elements of each stack, the stacks themselves, are formed based on performing well together.

More specifically, the program that will be created from each stack will perform actions that wholly utilize each stack element, in ideally equal ways. To do so, I will compose a sequence of actions that run algorithms on the databases of each stack. This plan will provide a method of convergence with vastly different technologies, to use for meaningful comparison. A method of convergence in this case has to do with preventing outside factors from affecting results that were not directly as a result of the stack’s elements. By ensuring that is the case, the true representative nature of each stack’s speed can be analyzed. The actions mentioned previously include the following. All stacks’ websites have a clickable button that when clicked will send an API (Application Program Interface) to the respective back-end server. This request asks for the backend server to retrieve all data from the respective database. Once retrieving all the data asked for, the backend server will respond to the API request with the data by sending it back. Once the client retrieves the data it will redirect the webpage and render all of the data to the webpage Using this procedure will utilize each stack’s database wholly. As well as prove to be a robust testing method for measuring time complexity.

Beyond the time metric, scalability will be researched and culminate to a decision on which size applications suit each stack. The way scalability is measured in this experiment is by introducing levels to the database element of each stack. The levels include different amounts of data: 100, 1,000, 10,000, 100,000 elements. Using four distinct levels leads to distinguishable results relating directly to the scalability of the stacks.

3.1 How Results Were Measured

After running the read-all data operation for each web development stack, and for each database size (100, 1000, 10000, 100000), timings were recorded. Times are recorded via Google’s Lighthouse API

Each Stack-Timing recorded is made of four distinct sequences.

1. Times start when a button with the label “Select All” is clicked.

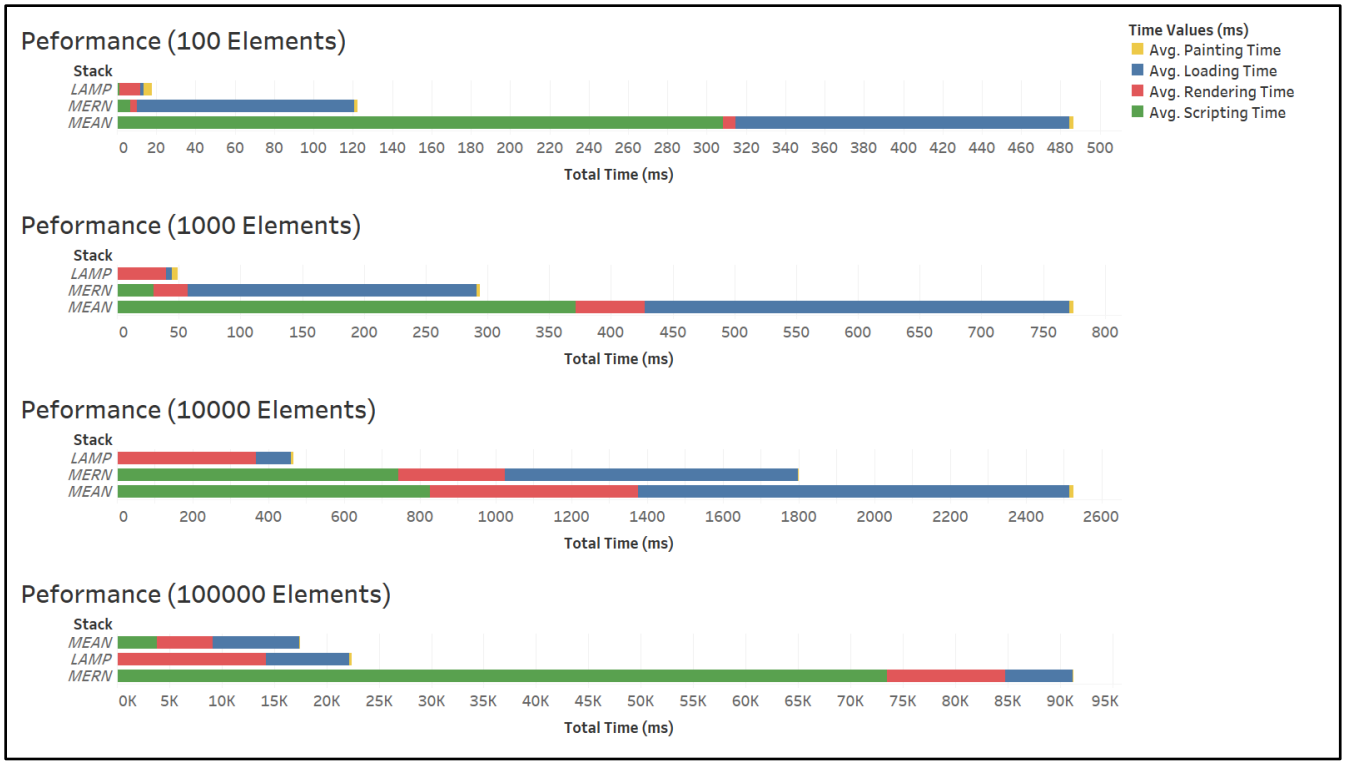

Figure 1. Average Time Performance per Stack per Scale

2. The frontend client sends an API request to the backend server to select all tuples/documents from the database [3].
3. The backend server retrieves the data from the database and sends it back to the client.
4. The client renders the data onto the page. This is where timing stops.

Total Time is calculated from the ten time audit measurements. Each audit is split into four parts. The ten time audits were summed together and averaged giving the Total Time metric. All times are measured and recorded in milliseconds (ms).

- Painting Time: Time it takes to paint all text/images on the page.
- Loading Time: Time spent parsing HTML, receiving data, completing network requests, receiving HTTP responses from a request, and sending network requests.
- Rendering Time: Time it takes to calculate document object model (DOM) styles.
- Scripting Time: Time spent executing javascript. [7]

3.2 Problems With Methodology

This research project's scope is limited to tests made solely on one operating system per stack. This scope limit leaves room for error that could not be directly tied to a stack's performance. The operating systems used were Kubuntu 20.04 and Windows 10. Although there are different operating systems used, that is one point of contention intentionally made in this project. Linux is one element of the LAMP stack, and must be compared against a non-linux operating system.

Another problem persists within using two different computers with different resource capabilities. To address this issue, we should've used virtual machines with specified resources for all stacks to use equally. This mistake added undetectable error.

4. RESULTS AND ANALYSIS

The results are significant for which stack performs better timewise at each database scale. It is clear for all practical scales. Although 100,000 data elements is very common, rendering 100,000 data elements on a webpage isn't practical, so it's important to mention that unless a webpage runs at a horribly inefficient extreme with its DOM element count, we can exclude the 100,000 elements scale from meaningful comparison.

The LAMP stack is most efficient time-wise. This stack retrieving and rendering the data quickest isn't surprising as frameworks such as React and Angular aren't likely to outperform basic scripting languages like PHP. The MERN stack outperforms the MEAN stack at each practical scale as well. This means we can safely order the stacks by performance clearly.
Stacks ranked by time performance.

1. LAMP
2. MERN
3. MEAN

An interesting anomaly or outlier from this data presented is that the MEAN stack had an overall faster average total time compared to both the LAMP and MERN stacks at the largest scale. This is

odd as the MEAN stack performed well below the other two stacks at every other scale. The reason for this is unclear. At 100,000 elements, MERN underperformed significantly. From online resources [5,11,12] it appears that React handles small payloads of data much faster than Angular, so it makes sense that at lower scales, the MERN stack is much faster.

One meaningful conclusion I want to include is that the metrics being compared in this project are not fully conducive to deciding whether you should pick one stack over another. However, It does provide a significant means of comparison for which development stack performs all tasks quicker. However, since the scope of this project is limited to database fetching and rendering of data, this might not provide an end all be all. As the LAMP stack might not always perform fastest in all use cases.

4.1 Testimony

Disclaimer: Unless otherwise noted, this is not meant to be scientific, rather a testimony of personal experience. While designing, organizing, composing, and testing this research experiment, I had a lot of time to learn, relearn, and enhance prior knowledge on all of the stacks elements. Take these factors into consideration while reading this section.

4.1.1 Composition

The core component of creating this project involved composing the stacks. This involved creating each stack, composing and integrating all four elements together. This comprised a majority of the time building the project. It is worth mentioning the ease of composition for each stack, since quick deployment is very important to many organizations and developers worldwide. I want to state that from personal experience, the LAMP stack was the easiest to compose. This is most likely due to the lack of connecting a framework (like React/Angular) to the other elements of the stack and the overhead required by frameworks to do so. In regards to the other two stacks, they were equally as difficult, but more so than LAMP, to compose as default entities.

4.1.2 Ease of Use

It is important to developers how easy software is to use. From my experience, the MERN stack was easiest to use, mostly due to the flexibility of the ReactJS framework. Although the LAMP stack was a close second. However Angular in the MEAN stack was horribly difficult to use, consistently, relative to the others.

5. CONCLUSION

Overall there is a lot gained from this research. As previously shown in other research, time performance for languages with no frameworks is much better than with. The results in this experiment reinforce that notion. It is evident the MERN stack is faster than the MEAN stack at managing and rendering data. This will prove useful for anyone looking to squeeze the most performance they can out of their applications. The LAMP stack is the choice for speed, as well as for developers who don't wish to use a framework within their stack. The MERN stack is for developers who are willing to sacrifice some speed to include the utility of a framework within their stack of choice.

5.1 Research Continuation

There are several ways this research project could be altered and continued.

5.1.1 Cloud Testing

With this project all experimental tests have been run locally. A follow up would be to run the same time tests for these stacks when they are deployed on cloud platforms such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud. This type of experiment will useful to larger projects and businesses that provide all of their services through cloud platforms. Depending on how each cloud platform integrates itself with each element of the stacks imposed on it, that could drastically alter results.

5.1.2 Database Scale

Another extension on this project that could be implemented is one of scale improvement. This scale improvement includes increasing the overall database size to scales of millions. This would be useful as the stack might converge or diverge at a higher scale, since we tested a limited subset of realistic database scales.

5.1.3 Statistic Analysis

There isn't currently any form of statistical analysis presented from this research project. What could be provided next are tests for significant differences between the times presented in the results section. This would add extra credence for choosing a stack over another based on speed.

6. REFERENCES

[1] "AngularJS." *Wikipedia*, Wikimedia Foundation, 21 Jan. 2021, en.wikipedia.org/wiki/AngularJS.

[2] "Database." *Encyclopædia Britannica*, Encyclopædia Britannica, Inc., www.britannica.com/technology/database.

[3] "Front End and Back End." *Wikipedia*, Wikimedia Foundation, 17 Jan. 2021, en.wikipedia.org/wiki/Front_end_and_back_end#Front-end_focused.

[4] Group, Documentation. "Essentials¶." *Welcome! - The Apache HTTP Server Project*, httpd.apache.org/.

[5] Jelisejevs, Pavels. "React vs Angular: An In-Depth Comparison." *SitePoint*, SitePoint, 24 Aug. 2020, www.sitepoint.com/react-vs-angular/.

[6] "Linux." *Wikipedia*, Wikimedia Foundation, 3 May 2021, en.wikipedia.org/wiki/Linux.

[7] Meggin KearneyFlavio Copes. "Timeline Event Reference." *Chrome Developers*, developer.chrome.com/docs/devtools/evaluate-performance/performance-reference/#scripting-events.

[8] "Node.js Web Application Framework." *Express*, expressjs.com/.

[9] Node.js. "About." *Node.js*, nodejs.org/en/about/.

[10] "React – A JavaScript Library for Building User Interfaces." *– A JavaScript Library for Building User Interfaces*, reactjs.org/.

[11] Systango. "Best Stacks For Web Development." *Medium*, Medium, 11 Jan. 2019, systango.medium.com/best-stacks-for-web-development-991f91b7f99c.

[12] "Top 6 Tech Stacks That Reign Software Development in 2020: Fingent Blog." *Fingent Technology*, 8 Dec. 2020, www.fingent.com/blog/top-6-tech-stacks-that-reign-software-development-in-2020/.

[13] "What Is MongoDB?" *MongoDB*, www.mongodb.com/what-is-mongodb.

Comparing the Effectiveness of Diegetic vs Non-Diegetic Interface Designs for 3D Manipulation in Virtual Reality

Abdullah Choudhry
Dr. Zhang Mingrui, Dr. Sudharsan Iyengar
Department of Computer Science, Winona State University, 175 W Mark St, Winona, MN 55987, U.S.A
Email: achoudhry17@winona.edu

ABSTRACT

Virtual Reality is becoming more and more prevalent in many domains. Along with this, VR technology has become increasingly advanced. VR Interfaces, however, still require further research to determine how they impact user experience and presence in virtual environments. In this paper, we explore the time-efficiency and overall "presence" of two UI patterns in a virtual environment, Diegetic and Non-Diegetic UI, for completing tasks related to 3D manipulation in order to determine which is the more effective form of UI for said tasks. We use a between-participants design and ask participants to complete specific tasks using the two UIs, measuring the time taken for each task and then having them take a brief presence questionnaire. Although we were unable to prove interface design has an impact on the effectiveness for 3D manipulation, we gained valuable insight into designing studies for 3D interface designs.

CCS Concepts

• **Human-centered computing → Virtual reality.**

Keywords

Virtual reality; 3D Manipulation; Diegetic; Non-Diegetic; Interface Design and Evaluation.

1. INTRODUCTION

In recent years, the use of Virtual Reality (VR) in various applications has increased extensively. As its importance increases in various domains such as science, education, and entertainment, we must look at the way we experience the tools and applications built in VR. Given the growing use of VR systems for applications in these domains and the improvements to designs of headsets and controllers, research into how 3D user interfaces designs affect user experience and behavior has become vital [6].

An essential characteristic of VR is the ability to create a sense of presence, or "being there" in a virtual environment [8] and depending on how a virtual environment is developed, the user experience can vary greatly. The main interfaces that users interact with can make or break the sense of presence and whether a VR application is usable at all. For example, Son Y et al. [6] found that when comparing hand and toggle interfaces for Virtual Reality-based Learning Environments (VRELs), that hand interfaces provided a marginally higher environmental presence over toggle environments. Additionally, Tanaka et al. [3] found that alternative interfaces for locomotion in VR reduced motion sickness in users significantly over an analog-like stick on a gamepad device.

For this study, we aim to look at two UI implementation patterns, Diegetic and Non-Diegetic, and their effectiveness for completing 3D manipulation tasks, or acts that involve physically handling objects [1]. More specifically, we hope to look at the time-efficiency and presence for completing tasks using these interfaces. Diegetic UIs are interfaces that exist within the environment, some examples are a watch that tells the time, or a compass that gives direction [7]. Non-Diegetic UIs are objects that do not exist within the environment. They are not part of the 3D space and have no depth. These are things like Heads-Up Displays or menus [2].

In order to test this, our plan is to have users complete various 3D manipulation tasks including positioning, rotation, or scaling. Simply put, these tasks involve the manipulation of objects in a virtual space while maintaining their original shape. An example of a 3D manipulation task could be positioning a medical probe relative to virtual models of internal organs in a VR medical training application [1]. We will record the time it takes for participants to familiarize themselves with the interface and complete the given tasks and then have them complete a presence questionnaire. Using this information, we hope to assess the effectiveness of the two interface designs and provide insight into what interface is more fit for completing 3D manipulation inside of Virtual Reality-based applications.

Proceedings of the 21st Winona Computer Science Undergraduate Research Seminar, April 19, 2021, Winona, MN, US.

(Hypothesis) VR applications that utilize diegetic interfaces are more efficient than non-diegetic, based on time-efficiency and presence, for completing 3D manipulation tasks.

2. METHODS

2.1 Participants

For this study, eighteen volunteer participants were recruited by email and through events among students at Winona State University. The participants included in the study were aged between 18-24 years. The study does not include people who are prone to motion sickness; struggle to balance, have a visual impairment, or hearing disability. Prior to the experiment, a self-survey was given to the participants. Data including gender, height, prior experience with VR, frequency of playing video games in the last 3 years were recorded. These questions were chosen based on the study done by Son Y et al. [6] on 3D interfaces designs for virtual learning environments.

Table 1. Participant demographics

Gender	Male = 13, Female = 5
Height(cm)	M = 175.97, SD = 12.77
Prior Experience in VR	2 had no prior experience of VR 11 had very minimum experience of VR 5 had a lot of VR experience
Video Game Experience in the Last Three Years	3 played 1-2 times a year 2 played 1-2 times a month 1 played weekly 12 played daily

2.2 Experimental Design & Procedure

A between-participants design was used for this experiment, where participants were put into one of two groups with Diegetic or Non-Diegetic. Participants then took the self-survey. Once they completed the survey, they put on the virtual headset and controllers and we positioned them into the virtual environment [6]. Inside of the virtual environment, the participants were given instructions to complete four tasks one by one. The tasks were as follows:

- Change the size of a cube to match a semi-transparent cube placed next to it [5].
- Rotate a building block to match a semi-transparent building block placed next to it [5].
- Scale a cylinder to fit properly in a nearby hole, then position it inside of the hole.
- Scale and rotate a building block to match a semi-transparent building block placed next to it.

These tasks were decided after researching 3D manipulation in VR [1]. For the diegetic interface, controls were integrated onto the table and a part of the environment, while for the non-diegetic interface, controls were displayed on a rectangular canvas in front of the camera.

Figure 1. Diegetic(left) and Non-Diegetic(right) interfaces used for the experiment

2.3 Measures

Completion Time. The participant's view inside of the virtual environment was recorded as they completed the experiment. Using this recording, task completion time was measured in seconds and calculated from the moment the participant pressed the start button until the time they took to finish all 4 tasks. Individual times for each task, from when they started it to how long it took them to press the finish button, were recorded as well.

Presence. After the VR-experiment was completed, participants were given an 18-item presence questionnaire adapted from previous literature [1,4,9]. This questionnaire is composed of questions in 3 main categories: involved/control, natural, and interface quality.

2.4 Apparatus

Participants used an Oculus Rift head-mounted VR headset. Alongside this they wore a pair of touch controllers to interact with the virtual environment. The virtual environment itself was created using a 3D game engine, Unity 3D. Unity was chosen as it is the premiere software for developing virtual reality-based applications [7]. Resources found on the Unity Learn website were used to assist in development of the virtual environment. The Unity Asset Store and Poly by Google were used to help provide assets for designing and developing the environment, and the Interfaces were built using the UI toolkit for Unity. Open Broadcaster Service (OBS) was used to record the view of the participant in the virtual environment. The computer we used for the experiment has the following specifications:

- Main Memory: 16 GB RAM
- CPU: AMD Ryzen 5 3600 6-core processor, 3.60 GHz
- GPU: Nvidia GeForce RTX 2060 SUPER

3.RESULTS

The data for this experiment was recorded into a Microsoft Excel spreadsheet and analysis was performed using JMP Pro 15, a statistical software. Normality of data was checked using the Shapiro-Wilk test [6]. Data is formatted as mean ±

standard deviation, unless otherwise stated. Significance criteria were set at p = 0.05.

3.1 Task Completion Time

There was no significant difference between the completion times for the two interface design conditions. [p = 1.0]. In fact, the average time for both conditions came out to be the same. After seeing this, it was decided that further investigation into the completion times for each task should be done. Looking at each specific task, there was no significant difference between the completion times of the two interface designs for any of the tasks. Task 1 p = 0.99, task 2 p=0.82, task 3 p=0.6274, task 4 p=0.33.

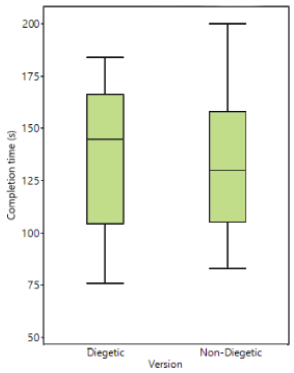

Figure 2. Plot of completion times

Table 2. Summary statistics for completion time (total & per task)

Dependent Var.	Diegetic	Non-Diegetic
Total Completion(s)	138.89 ± 36.30	138.89 ± 36.24
Task 1 Completion (s)	33.67 ± 16.73	33.56 ± 15.10
Task 2 Completion (s)	20.56 ± 15.97	21.89 ± 4.73
Task 3 Completion (s)	46.67 ± 22.28	42 ± 17.37
Task 4 Completion (s)	33 ± 12.24	39.78 ± 16.56

3.2 Environmental Presence

There was no significant difference between the measured environmental presence for the two interface design conditions [p=0.69]. However, the Non-Diegetic version had a marginally higher average score compared to the Diegetic Version (Table 3). Further investigation into scores for the individual categories (involved/control, natural, interface quality) showed no significant differences for the two interface design conditions. Involved/Control p = 0.57, Natural p = 0.34, Interface p = 0.52.

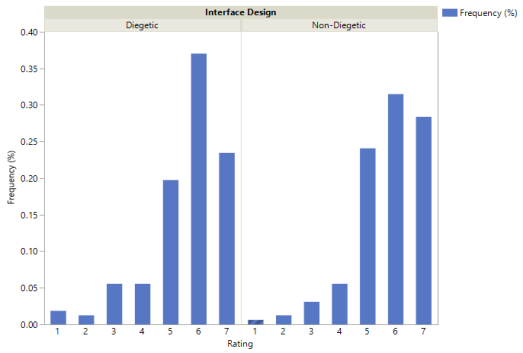

Figure 3. Overall frequency of ratings from the presence questionnaire

Table 3. Summary statistics for environmental presence (total & per category)

Score Type	Diegetic	Non-Diegetic
Overall Presence Score	95.11 ± 18.98	97.67± 7.42
Involved/Control score	34.44 ± 6.40	34.11± 4.0
Natural score	39 ± 7.45	40.11 ± 3.04
Interface score	21.67 ± 4.30	23.44± 2.35

4. DISCUSSION

4.1 Hypothesis

Results of the study do not support the proposed hypothesis. There was no significant difference between the two interface designs on the dependent variables. In fact, the average completion time had no difference at all between the two interface designs. Neither completion time nor environmental presence were significantly impacted by the interface design.

4.2 Limitations

This study had several limitations. First, the study sample size was fairly small (N = 18). After separating the participants into groups, there were only 9 participants for each condition. There was also not an even split for gender, and the background of the participants was not diverse. Many participants were white male students in the Computer Science department at Winona State University.

Second, more research into how to measure presence needs to be done. Presence is a complex measure with multiple aspects. The main topics in the post-test questionnaire were inconsistent in depth, and certain aspects of presence had more weight on the final score. In previous works post-test questionnaires are the most frequently used measure of presence. As stated by Schwind et al. however, this method relies on the subject's memories of the VR experience. These memories are often inconsistent or offer an incomplete

picture of the VR experience the participant had. Ideally another measure of immersion in real time inside of the virtual environment should be used [8].

Third, there was too much variance in user input for the two versions of the application. User input was handled differently for the Non-Diegetic application in an attempt to make it seem less a part of the environment, however there is a distinction between user input and user interface so the method for user input should have been handled the same for both versions.

On the topic of user input, we observed that many participants instinctively tried to grab the objects that they were meant to perform tasks on. This feature, however, was not enabled inside of the environment. The environment should be changed so that users don't feel the need to grab the objects or the feature should be added to the application.

Fourth, the tasks in the environment were too simple, or there were not enough tasks. In either case, the completion times were too short for a valuable result. More complex and nuanced tasks that push the limits of the two interfaces would help to make this measure more meaningful and could reveal stronger relations between the conditions and completion time.

Along with the tasks being too simple, the scenario created in the virtual environment was not very believable. More knowledge and experience in building applications for VR would allow us to create an environment with tasks that mirror something one could see in real life, providing an experience that feels more natural and intuitive to the user.

Fifth, in the application created for the study, participants sat in a stationary position. Most VR applications, however, require users to stand inside the virtual environment. It should also be noted that during the study, many participants with lower heights struggled with vision as they had to look up at more than other participants, and some participants decided to stand in order to complete the tasks.

Sixth, while task completion time is one measure of performance, a more holistic assessment of the quality of the controls for completing 3D manipulation tasks is required [6].

4.3 Conclusion

In this study, we were unable to prove that interface design has a significant impact on completing 3D manipulation tasks in VR. While our hypothesis was proven wrong, we gained insight on how to improve methods of testing 3D interface designs for 3D manipulation tasks. In the future, a larger, more diverse, sample size of participants is required. The virtual environment and related tasks need to be more complex and realistic to allow for more valuable measurements. More work needs to be done to measure presence in real time while still holding the participants immersion.

5. REFERENCES

[1] Bernhard E. Riecke, Joseph J. LaViola Jr., Ernst Kruijff. 2018. 3D user interfaces for virtual reality and games: 3D selection, manipulation, and spatial navigation. dl.acm.org/doi/10.1145/3214834.3214869

[2] Brongo, Roxana. "How to Design the Best UI for Room-Scale VR." *Valtech.com*, Valtech, 18 Apr. 2017, www.valtech.com/insights/how-to-design-the-best--ui-for-room-scale-vr/.

[3] Eduardo H. Tanaka, Juliana A. Paludo, Leonardo R. Domingues, Carlúcio S. Cordeiro, Olavo Giraldi, Marcos H. Cascone, Edgar V. Gadbem, and Adriana Euflasino. 2015. User interface design of an immersive virtual reality environment to electricians training. In Proceedings of the 14th Brazilian Symposium on Human Factors in Computing Systems (IHC '15). Association for Computing Machinery, New York, NY, USA, Article 36, 1–10. DOI:10.1145/3148456.3148492

[4] Katy Tcha-Tokey, Emilie Loup-Escande, Olivier Christmann, and Simon Richir. 2016. A questionnaire to measure the user experience in immersive virtual environments. In Proceedings of the 2016 Virtual Reality International Conference (VRIC '16). Association for Computing Machinery, New York, NY, USA, Article 19, 1–5. DOI:10.1145/2927929.2927955

[5] Robin Schlünsen, Oscar Ariza, and Frank Steinicke. 2019. A VR Study on Freehand vs. Widgets for 3D Manipulation Tasks. In Proceedings of Mensch und Computer 2019 (MuC'19). Association for Computing Machinery, New York, NY, USA, 223–233. DOI:10.1145/3340764.3340791

[6] Sun Y, Kar G, Stevenson Won A, Hedge A. Postural Risks and User Experience of 3D Interface Designs for Virtual Reality-based Learning Environments. Proceedings of the Human Factors and Ergonomics Society Annual Meeting. 2019;63(1):2313-2317. doi:10.1177/1071181319631023

[7] Unity Technologies. "VR Best Practice." *Unity Learn*, 13 Feb. 2020, learn.unity.com/tutorial/vr-best-practice.

[8] Valentin Schwind, Pascal Knierim, Nico Haas, and Niels Henze. 2019. Using Presence Questionnaires in Virtual Reality. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). Association for Computing Machinery, New York, NY, USA, Paper 360, 1–12. DOI:10.1145/3290605.3300590

[9] Whelan, T.. "Social Presence in Multi-User Virtual Environments : A Review and Measurement Framework for Organizational Research." (2008)

How Different Programming Languages Are Used In Cross Site Scripting Attacks

Alex Feller
Advisors: Dr. Mingrui Zhang, Dr. Sudharsan Iyengar
Computer science department, Winona State University
Email: afeller16@winona.edu

Abstract

JavaScript and PHP are two very popular languages in the world of coding, especially for websites and web-based programming. This paper will be studying the effectiveness of JavaScript in preventing web-based cyber-attacks, and its roles/use in web-based code. This paper will also examine Cross Site Scripting attacks and its gaining popularity. The client side is one of the vulnerable aspects for cyber-attacks, solutions are available to prevent such attacks against web applications, and we will compare JavaScript with php with respect Cross Site Scripting Attacks.

1. Introduction

In order to begin a career in the field of cyber security, one would think some basic coding skills may be required to be qualified for these types of jobs. It would be unrealistic to learn every language in your college career and/or free time, so there is a need to know which language or languages will be the most helpful in a field that one intends to enter. The investigation of what language would be most important to know is helpful because students
wanting to enter the field of cyber security, will be able to use this information to familiarize themselves with the best

language to know. According to Keith S. Jones, Akbar Siami Namin, and Miriam E. Armstong, they agree that the need for cyber specialists is increasing, but ranked JavaScript in the top 5 of their list for languages to learn and know. “The 5 most frequently listed languages accounted for 69% of responses and had a mean importance rating of 4.36 (SD = 1.50): Python (N = 29, M = 4.5, SD = 1.57), languages from the C family (N = 19, M = 4.68, SD = 1.34), JavaScript (N = 17)...”

The specific attacks in this paper relate to a popular attack known as a Cross Site scripting attack (hereafter referred to as xss). There are several outcomes from a successful attack on a website, whether it be a consumer’s information or the owners. A user’s device can be compromised and used in a botnet, which then can be used for other malicious purposes. Personal information stored on a device can be at risk and allow the attacker to spy on a user’s network and personal use for their device, or alternatively an attacker could completely ruin/disable a user’s device. Generally xss attacks are used to target user information and steal credentials

Php and JavaScript are at the top of the list of languages to know for cybersecurity, Based on the gathering of this list of languages, I will conduct research to support my hypothesis that JavaScript is important to know in order to potentially defend against xss attacks, or conduct your own in order to find whether or not your own website may be vulnerable to this kind of attack.

CCS Concepts

Web security → Cross site scripting attacks

Keywords

Web-attacks, cross site scripting attacks, JavaScript, php, programming languages.

2. XSS Attacks

Recent data from SANS institute estimates that up to 60% of Internet attacks target web applications. These attacks are often successful because developers may have little to no security background. The first thing to do is identify the two

main languages in attacks and XSS, which is php (Hypertext Preprocessor) and JavaScript. To better understand the attacks the language used should be broken down. First, we'll look at php. Php is a server-side scripting language . PHP is often used in websites involving user inputs, it allows for a wider array of things to be done and more control over aspects of the input, but it provides a new opportunity for exploits. Server side is important because Although XSS indicates an attack against the client-side's web browser, exploitation of its abilities occurs on the web server side, which is what php is designed for. Most commonly bypassing php is known as not sanitizing input data, which will allow a user to exploit an input and then execute their own code within the website, and control what a visitor may see or do, such as logging their personal information, or displaying a false page to trick users into entering their credentials for what they think is a trusted site. Generally, XSS attacks are used to steal user information. These kinds of attacks will often inject JavaScript into the client side of an application, in order to execute malicious code without the user or owner of the web application knowing about it. There are several things that could be disastrous that are done using these attacks such as distributing malware to users on the site, data theft or even remote control of the application. Xss attacks can heavily rely on character inputs that don't match the description field of the input, but the characters still exist in the source code, the browser sees this input as part of the page and then can allow a script to run that is not supposed to.

2.1 Reflected xss attacks

There are three main types of xss attacks, the first and simplest of the 3, is a reflected xss attack (also known as non-persistent). This occurs when the web application receives data in a request and includes the data in the response from the browser immediately, for example. The application has a search function that a user inputs a search parameter, the browser then echoes the search without processing the data, which could allow a script to run in between, and echo back the search in an unsafe way and executes a potentially malicious script.

2.2 Stored-XSS

The second type of xss attacks is known as a stored xss attack (also known as persistent) that occurs when a script is injected directly into a web application, receiving data from an untrusted source and injected into the application. Generally this is JavaScript, and can be potentially stored in different places depending on the manner of the site, and what input fields are available

2.3. DOM-XSS

The third most common type of xss attack is known as DOM-based xss, this is a more advanced and sophisticated type of xss attack, this involves modifying the DOM (document object model) environment. This is generally a client-side attack, and malicious code never needs to be sent to the server, just executed in the victim's browser using the original script in the web application.

3. JavaScript

JavaScript injection and XSS attacks are not the same but they go hand in hand, injecting malicious JavaScript is how an attacker takes advantage of security flaws in a webpage and can use their own script to take information. JavaScript allows an attacker to execute arbitrary commands and display arbitrary content in a user's browser, and further manipulates it from there. Despite these languages being at the center of xss attacks, they can be used to help prevent them as well. Libraries in PHP and Java exist specifically to help sanitize input that are maintained and frequently updated, this will be discussed in the prevention and solutions section of this paper. The best defense of fending off attacks is being aware of how JavaScript can be executed and being aware of programming errors that may cause these to occur. JavaScript. The overall intent of an xss attacks is to return malicious script back to a user, there are other scripting languages that can be used to execute an xss attacks, however because JavaScript is nearly fundamental to a successful web application a user will find it in almost in almost every website, whereas other scripting languages may be found in web applications but not as frequent, nor are they as popular, also making it less likely that an attacker may choose to use a different attackers script that they have used to successful execute an xss attack.

3.1 Php's role in XSS

Php is another popular scripting language for web development that can be implemented into html. Unlike JavaScript, php is a server-side scripting language, but still plays an important role in a web application that chooses to utilize it in their site. One of the most common ways an xss attack is executed is through web forms or link using a client or server-side script such as php, which php is very often used for.

4. Solutions and prevention

There are still plausible solutions and prevention actions that can be taken in order to make sure a web application is not vulnerable to an xss attack. One thing one could do is enable an xss filter, php and html both have libraries that claim to filter xss attacks. Sanitizing any user input is also very important and can prevent the possibility of certain xss attacks before they can even happen. There are several modulators that exist for JavaScript that can sanitize input and validate user input, two of the most popular ones are 'validator.js' and 'yup', there exists more libraries besides these two that can also be used to sanitize input, and therefore eliminate the possibility of an xss attack occurring. For any input fields that rely on a php script rather than a JavaScript script, htmlspecialchars() is going to be very useful. What this tool does is that anything imputed with no recognized or blank characters will convert them in order for the browser to recognize them.

5. Conclusion and Analysis

After breaking down the different methods of xss attacks, and the roles of the two main scripting languages used in these attacks, I back my hypothesis that JavaScript will be a better language to learn and understand in order to perform an effective static analysis of a web application. Along with the tools that allow developers to help secure their site using JavaScript, the scholarly papers that reviewed also reflect that JavaScript shows up far more in the three main kinds of xss attacks, and is one of the more preferred languages for the client side end of websites.

References

[1] Rodríguez, Germán E et al. "Cross-Site Scripting (XSS) Attacks and Mitigation: A Survey." Computer networks (Amsterdam, Netherlands : 1999) 166 (2020): 106960–. Web.

[2] Sarmah, Upasana, D.K Bhattacharyya, and J.K Kalita. "A Survey of Detection Methods for XSS Attacks." *Journal of network and computer applications* 118 (2018): 113–143. Web.

[3] Fraiwan, Mohammad et al. "Analysis and Identification of Malicious JavaScript Code." *Information security journal.* 21.1 (2012): 1–11. Web

[4] Scholte, Theodoor, Davide Balzarotti, and Engin Kirda. "Have Things Changed Now? An Empirical Study on Input Validation Vulnerabilities in Web Applications." *Computers & 8security* 31.3 (2012): 344–356. Web.

[5] Gupta, Shashank, and B.B Gupta. "Automated Discovery of JavaScript Code Injection Attacks in PHP Web Applications." Procedia computer science 78 (2016): 82–87. Web.

[6] Nicula, Ștefan, and Răzvan Daniel Zota. "Exploiting Stack-Based Buffer Overflow Using Modern Day Techniques." *Procedia computer science* 160 (2019): 9–14. Web.

[7] Malviya, Vikas K, Sawan Rai, and Atul Gupta. "Development of Web Browser Prototype with Embedded Classification Capability for Mitigating Cross-Site Scripting Attacks." Applied soft computing 102 (2021): 106873–. Web.

[8] Marashdih, Abdalla Wasef et al. "Web Application Security: An Investigation on Static Analysis with Other Algorithms to Detect Cross Site Scripting." Procedia computer science 161 (2019): 1173–1181. Web.

[9] Hydara, Isatou et al. "Current State of Research on Cross-Site Scripting (XSS) – A Systematic Literature Review." Information and software technology 58 (2015): 170–186. Web.

[10] Kirda, Engin et al. "Client-Side Cross-Site Scripting Protection." Computers & security 28.7 (2009): 592–604. Web.

[11] Li, Xiaowei, and Yuan Xue. "A Survey on Server-Side Approaches to Securing Web Applications." ACM computing surveys 46.4 (2014): 1–29. Web.

Using Support Vector Machine Learning to Predict Game Entity in a "Super Smash Bros. Melee" game

Trevor Firl

Winona State University
175 W Mark St.
Winona, MN 55987
+1 (507) 457-5000

trevor.firl@gmail.com

ABSTRACT

Over the years fighting games have gotten more refined as in-game character mechanics have grown significantly more complex. With software advancements, players of the popular platform fighting game *Super Smash Bros. Melee* (SSBM) are able to save replay files of matches and extract metadata from previously played matches. Information from matches can be used to train classification models to predict aspects of the game such as the character played. With a diverse cast of characters to choose from, each character has a unique move set to use during a match. However with the complex nature of the SSBM in-game environment on top of the multi-player element of the game, the ways in which a character can be controlled is nearly limitless. With a refined metagame of many characters, but the dynamic ability of the game environment, this poses as a clash between the predictability of certain characters and the mind games of the human player. Nonetheless, this project aims to show the predictability in different characters within a SSBM match, using information and metadata extracted from previous replay files of matches to train a Support Vector Machine learning model.

1. INTRODUCTION

Super Smash Bros. Melee (SSBM), is a popular platform fighting game released in 2001 for the Nintendo GameCube. Shortly after release, a competitive scene emerged as people attended tournaments and trained to become better with their character of choice. With no online capabilities for the game on the original console, as well as no replay functionality, the only replays of tournament matches were recorded via external sources such as video cameras, capture cards, or live streaming. However, these methods give no real numerical analysis about matches being played. Come June 2018, Jas 'Fizzi' Laferriere with the help of others release *Project Slippi*, an open source project with the goal of creating an easy to obtain, data-rich replay file of SSBM matches. The goal of this project is to analyze the game data including controller inputs, action states, metadata, and more [1]. With the help game emulators capable of online play, this project would go on to be used in online matchmaking for everyone to use, and to extract a replay file capable of being analyzed in real time or after a match. With replay files available, further analysis can be done on real matches played to better understand how humans play any SSBM match.

Over the 19 years since SSBM's release, the way in which a character is played within the game has become refined. Some moves may be more useful than others for certain playable characters, and are used more often. Every character has unique combos and strategies done using unique controller inputs. However, despite this, the game is still vastly free-form, and has many other factors that come in to play when playing any given character. Humans may play their character differently depending on opponent's character, the in-game stage being played on, the skill level of the opponent, and more.

In this paper, we focus on whether a character played in a *Super Smash Bros. Melee* match is predictable using Support Vector Machine learning on metadata, statistics, and information derived from previously played SSBM matches.

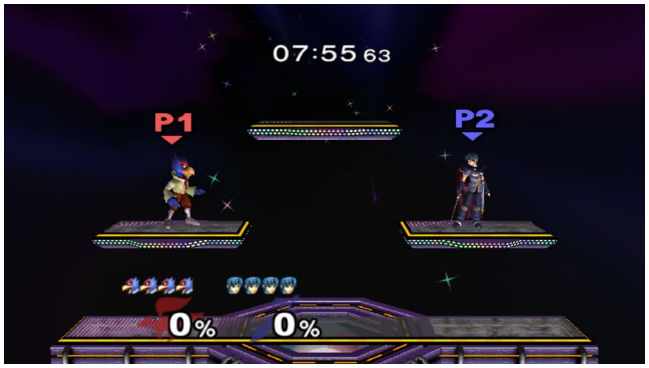

Figure 1. A typical match between two popular playable characters: Falco (left) and Marth (right) on the Battlefield stage

2. BACKGROUND RESEARCH

While individual sets played can now easily be analyzed using *Slippi*, these analyzed sets have yet to be compiled and analyzed on the broader scope of entire characters. Other strides of research have been done around this game including using deep learning to beat professional *Super Smash Bros. Melee* players at their own

Proceedings of the 16th Winona Computer Science Undergraduate Research Seminar, April 19, 2021, Winona, MN, US.

game [2]. There are also various other artificially intelligent bots, such as *SmashBot* which aims to beat human players as efficiently as possible using strategies, tactics, and chains as described by the creator [3]. In this regard, understanding a "best" way to play and teach a computer is achievable with this method. However, this project aims to analyze existing matches previously played in a tournament setting to learn how humans play different popular characters. By using multi-classification Support Vector Machine (SVM) learning on these played matches, a prediction of which character was played on one side of a Super Smash Bros. Melee match can be generated. This is done to further analyze human players rather than build artificial players such as *SmashBot*. With a database of over 95,000 replay files saved, metadata is parsed using the software package "slippi-js" to create training and testing data for the Support Vector Machine to use.

3. METHODS

In this section, the methodology of collecting the appropriate replay files will be covered, as well as an overview of the dataset. Data preparation and the decisions made for filtering and parsing files into an appropriate format using package software follows. Finally, the model used for training and testing data is covered.

3.1 Collecting Data

A public dataset of replay files comprised of real matches played between two players was be used as the source of training and testing data. This dataset was compiled by the SSBM community, and contains tournament sets from many groups of players around North America. This dataset is commercial use and free to use. Since the matches are from tournament settings running a best-of-3 and/or best-of-5 formats, multiple matches between the same players are present. From this information, it can also be deduced that players of higher skill may have more matches present in the dataset than a lower skilled player. Albeit, this also means there is a diverse set of matches between different skill levels amongst the dataset. Since each match is in a tournament set, it can also be assumed that there is incentive to win, and therefore neither player is purposefully trying to lose a match. The dataset includes a total of 95,102 replay files already pruned to remove fake or bad matches for the analysis. This pruning includes matches that lasted under 30 seconds, matches that do not have a complete recording due to outside factors such as power loss or faulty recording of the match, and matches with more than two players present.

3.2 Data Preparation

3.2.1 Choosing the Characters

To narrow the scope, matches only including 4 of the 26 playable characters in the game are included for the training and testing sets. Amongst the characters chosen are:

- **Fox**, a fast and technical character. Chosen based on popularity along with the similarities to Falco. 3841 instances present in the dataset.
- **Falco**, another fast, combo heavy character with a great projectile. Chosen based on popularity and similarity to Fox. 3087 instances present in the dataset.
- **Marth**, a sword swinging character capable of walling out an opponent. Chosen on popularity and differences to the other three chosen characters. 1843 instances.
- **Jigglypuff**, a floaty, generally slow and methodical character. Chosen for the unique playstyle. 495 instances.

Instances indicates amount of times information was extracted from that character as the character to predict. However, matches with each of these four characters as the opponent character are also present. These four characters are among the most popular, so they have some of the most data present. The four are also considered some of the best characters in the game based on public opinion and tournament results, which make them some of the most in-depth characters in the game as well based on character mechanics, as well as human experience playing them.

3.2.2 Choosing the Stage

To narrow the scope further, training and testing data will also only includes matches played on 1 of the 6 possible legal stages. The "Battlefield" stage was chosen because it is the most common neutral stage played on in this dataset. It is also generally seen as the most well rounded stage for most characters in the game.

3.2.3 Project Clippi

The matches were filtered with the use of the Project Clippi framework which is used to rename replay files based on match information, including characters and stage [4]. The files were named to include only the character names and stage name. These files were then filtered to only include matches with the four appropriate characters as well as the Battlefield stage. The narrowed scope resulted in a total of 9266 unique match files.

3.2.4 slippi-js

These filtered matches were parsed using slippi-js to parse the needed metadata into a CSV format for the SVM to use. *Slippi-js* gives all frame and state information about a match to parse yourself, but many statistics are pre-computed and made readily available in JavaScript objects. The meta-data and information extracted for use of features in this project include:

- Character Played (being predicted)
- Opponent Character (OC)
- Inputs per minute (IPM)
- Win/Loss of the match (W/L)
- Length of the match in frames (Length)
- Openings per kill (OPK)
- Damage per opening (DPO)
- Neutral win ratio (NWR)
- Opening conversion rate (OCR)
- Dash-dance count (DDC)
- Wave-dash count (WDC)
- Ledge grab count (LGC)
- Number of grabs (Grabs)

The statistics retrieved were based on the character being played (the character to be predicted). All but the number of grabs are pre-computed statistics.

In order to get the number of grabs feature, each frame of a match needs to be read. At the same time, the action state ID of the played character was compared to that of the action state of a grab on each frame of the match, and then counted if matching. The total number of grabs used in the game was then also included in the training and testing file.

The decision of including these features was based on what separates certain characters apart from others, which was inevitably subjective.

3.3 Training Model

Traditionally the SVM model is used for binary classification; however with approaches such as One-vs-Rest or One-vs-All (OVA) and One-vs-One (OVO), SVMs are able to split multi-class problems into one binary classification problem per class or per pair of classes. These two approaches are more frequently used in practice as there are multiple efficient software packages already used for binary classification problems [5].

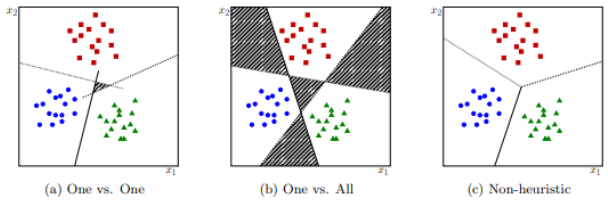

Figure 2. "Illustration of ambiguity regions for common heuristic multiclass SVMs. In the shaded regions ties occur for which no classification rule has been explicitly trained. Figure (c) corresponds to an SVM where all classes are considered simultaneously, which eliminates any possible ties." [6]

For this project, the Support Vector Classification (SVC) package was used from the scikit-learn software library. The multiclass support in this package is done using the One-vs-One scheme [7]. This library was chosen for its wide variety of high-level yet efficient supervised and unsupervised machine learning packages, making it accessible for non-specialists and academic work [6]. The Radial Basis Function kernel was used with a C value of 1 to indicate a higher penalty parameter, a common kernel and C value for SVM classification models. Since only multiple thousands of samples were used in this project, time scaling with SVC was not issue. 10-fold cross validation was done on the shuffled dataset to ensure the model is appropriately fit. Random samples of matches were used to create the training and testing data sets, with 75% being used for training, and 25% used for testing. After training, predictions were made on the testing data for an accuracy score. From here, precision scores for each class can be also be determined.

4. RESULTS AND ANALYSIS

With 10-fold cross validation, a mean accuracy of 72.81% was achieved. An accuracy of about 72.64% overall between the four different classes was achieved. Precision scores for each character class were the following:

Table 1. Prediction accuracy by character

Character	Accuracy
Fox	78.27%
Marth	52.62%
Falco	83.58%
Jigglypuff	39.55%

As we can see, Fox and Falco have higher accuracies compared to Marth and Jigglypuff. This may be due in part to the unbalanced data presented to the training algorithm. With Marth making up 19.9% of the instances in replay files, and Jigglypuff with only 5.3% of the instances. While Fox made up a larger 41.5% of the instances, and Falco making up 33.3%. With a normalized confusion matrix, we are better able to see the mistakes made by the algorithm.

4.1 Misclassifications

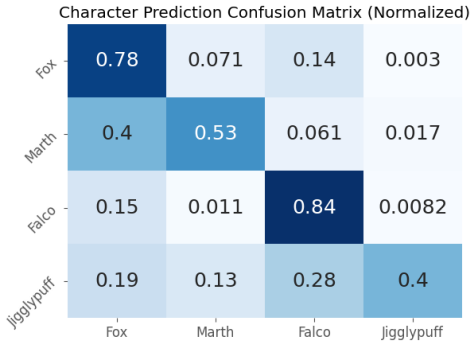

Figure 3. Normalized confusion matrix of the testing data

From Figure 3 we can what each character is being misclassified as the most. We can also try to give explanations as to why on a human level of playing.

4.1.1 Fox

Fox misclassified as Falco 14% of the time. This may be due to the similar fast playstyle generally seen in both of these characters.

4.1.2 Marth

Marth misclassified as Fox 40% of the time. This may be due to both characters great grabs, causing them both to grab more than other characters.

4.1.3 Falco

Falco misclassified as Fox 15% of the time. Similar to Fox, this may be due to the fast playstyles of both characters, as well as overlap in competitors who choose to play both characters.

4.1.4 Jigglypuff

Jigglypuff misclassified as Falco 28% of the time, Fox 19% of the time, and Marth 13% of the time.

4.2 Feature Importance

When classifying characters, some features may prove to be more important than others. While using the One-vs-One scheme, these features would be weighted based on the prediction of a certain character over another, rather than compared to the rest of the characters as would be seen in a One-vs-Rest approach. With the Radial Bias Function kernel used to compute accuracies as shown in Table 1, feature importance is not visible. Switching to the linear kernel allows us to peak at this One-vs-One approach. Albeit, at a lower accuracy than RBF, thus possibly skewing the actual importance of some features.

Figure 4. Feature Importance for Fox compared to Marth (Using the Linear Kernel)

From Figure 4 we can see there are four features helping the algorithm correctly predict Fox instead of Marth (see section 3.2.4 for abbreviation meanings). The best deciding factor being IPM. Fox is overall a fast, technical character, requiring higher amounts of button inputs to play at a competitive level. This can further our confidence that this feature is indeed classifying Fox the best compared to Marth. The number of grabs is the least helpful, and even a hindrance to predicting Fox over Marth. Both characters are seen as having good grabs, so the idea that this type of feature is not helpful to distinguish the two can be attributed to this.

5. CONCLUSION

Super Smash Bros. Melee is a fast paced multi-player platform fighting game with many complex character mechanics unique to different characters. Statistics can be extracted using software packages to analyze information about games previously played such as inputs per minute, length of the game, etc. While the differences and similarities in characters may show patterns in extracted statistics when given to a Support Vector Machine, there is always still human control of the character. This gives freedom to break patterns seen in certain characters, ultimately hindering the predictability to some extent. Nonetheless, with the use of Support Vector Machine multi-classification, we are able to show that these characters can be generalized and predicted based on game information to a moderate extent, showing more promise of high accuracy in certain characters such as Fox and Falco. For further research, proper feature extraction may lead to more promising results. To further improve accuracy, using bootstrap aggregation should be considered as well. On top of this, expanding the scope of the project to include matches on more stages to compare accuracy results on a stage by stage basis.

6. ACKNOWLEDGMENTS

Special thanks to Dr. Iyengar and Dr. Zhang for guiding me through the research process, and to Dr. Ma for helping me find an algorithm to work with. Of course, this project would not have been possible without Fizzi or the rest of the Project Slippi team.

7. REFERENCES

[1] Laferriere, Jas. “Project Slippi Public Release.” Medium, 17 June 2018, medium.com/project-slippi/project-public-release-4080c81d7205.

[2] Firoiu, Vlad et al. “Beating the World's Best at Super Smash Bros. with Deep Reinforcement Learning.” arXiv:1702.06230 (2017): n. pag.

[3] AltF4. “SmashBot.” Github, 1.0, 1 Nov. 2015, github.com/altf4/SmashBot.

[4] Au, Vince. “Project Clippi.” GitHub, 1.5.2, 30 Sept. 2019, github.com/vinceau/project-clippi/blob/master/README.md.

[5] Gerrit J. J. Van Den Burg and Patrick J. F. Groenen. 2016. GenSVM: a generalized multiclass support vector machine. J. Mach. Learn. Res. 17, 1 (January 2016), 7964–8005.

[6] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, null (2/1/2011), 2825–2830.

[7] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Infiltrating Cloud Storage of IoT Devices Using Ransomware

Anna Millerhagen
411 w 8th st winona 55987
(612) 875-5466
Amillerhagen16@winona.edu

ABSTRACT

Security is necessary for all areas of computer science. The expanding world of IT is IoT devices. There are many smart devices in our daily lives such as smart speakers, smart light bulbs, smart watches, doorbell cams, security systems, smart smoke alarms, smart cars, and many more. The need for security in these devices is critical. Any one of these devices could be the weak link to a security breach. These devices are all enabled and communicate through cloud services. They interact with various devices from different vendors all operating to provide the user with the best possible experience. The cloud authentication between devices could lead to a possible inflatable vulnerability. This paper explores the possible weakness and seeks to exploit them to understand the how to better prevent the attacks in the future. The aim of this paper is to infiltrate a device with known security weaknesses and access the cloud through the weak device. Then the final process would be to access a more secure device that holds more user data through the previously infiltrated cloud. This process proved unsuccessful.

Keywords

IoT, Ransomware, Penetration Testing, Zigbee, Wi-Fi

1. INTRODUCTION

The philosophy of many security people is that security begins when developing the object and is an important part of the developing process, security never ends. Many aspects of computer science are aware of this rule and implement security through their production and deployment of their technology. IoT or internet of things devices are a new and exciting realm of computer science. However, has the standard practice of security been utilized in the IoT devices that we use in our daily lives.

IoT devices range from simple Bluetooth enabled headphones to smart TVs. Anything that can be improved by adding a computer chip can become an IoT device. These devices are meant to improve life. One example is a home security system created by Smart cameras that communicate with each other and your phone to send an alert when an intruder is present. Another occurrence is smart speakers linked to all play the same song in different rooms, or cute dog treat dispensers that use a camera and your phone to know when to dispense a treat. IoT devices are present in cars to learn driving habits. However, many of these devices are not created using the best security practices. Many have password hardcoded into the device, and others have security implemented seemingly as an afterthought.

Proceedings of the 21thWinona Computer Science Undergraduate Research Seminar, April 19th, 2021, Winona, MN, US.

IoT devices can be connected, and their information stored through many methods. They are often connected to a phone or smart speaker through Bluetooth, then communication is sent through the BLE, IEEE 802.15 or the IEEE 802.11 communication protocol. IEEE 802.15, also known as Zigbee, is a protocol to create personal area networks with low-power digital radios. IEEE 802.11, or Wi-Fi, is a local area network, lan, and is generally what people use when connecting to the internet. BLE, Bluetooth Low Energy, is also a wireless personal area network based off Bluetooth. Their information could be stored within the device or on separate microchips that the user must input. It can Also be stored virtually in a cloud.

Some IoT devices are connected to cloud services to save the data that is on them. “These clouds are operated by both device vendors (Philips Hue, LIFX, Tuya, etc.) and cloud providers (Google, Amazon, IFTTT, etc.), offering integrated services for IoT users to control their devices across the Internet in a convenient and transparent way” [1]. Often a device is connected to multiple clouds that are all separate vendors. For example, a Google Home device is connected to googles cloud service to store the information received by the device. A Google Home will also be connected to many different devices from many different vendors. These separate devices connect to their vendors cloud services as well as googles.

“Ransomware is a form of malware that denies victims access to their resources until a payment is made” [2]. It can prevent users from using the device the ransomware is present on and could also prevent users from accessing data. Malware is the broader category of malicious software that ransomware is a part of. Ransomware can and has been used to prevent users from accessing their devices and data. Currently Ransomware is seen as more lucrative when used to attack large entities. “In 2019, the U.S. was hit by an unprecedented ransomware attacks that impacted at least 113 state and municipal governments and agencies, 764 health care

providers, and 89 universities, and 1233 schools" [5]. However, the presence of IoT devices may prove to be an enticing target.

The goal of this research is to prove that authentication and communication mechanisms of cloud storage from IOT devices are vulnerable to ransomware.

2. BACKGROUND RESEARCH

2.1 Arguments against Ransomware

IoT devices are small internet enabled devices that are meant to provide services to users. These devices have recently been the focus of malware users [2]. Ransomware has not yet been used to gain access to IoT devices on a large scale [2]. This is due a couple factors, firstly IoT devices have limited space and that can limit the effectiveness of ransomware. Ransomware works by bricking a device and demanding a ransom from the user before the user can gain access to the device again. This is ineffective in IoT devices because most IoT devices limit the amount of information that is written to main memory. This means that a user can reboot the IoT device and gain access to the device without paying the ransom [2]. Another reason that ransomware has not caught on with IoT exploiting criminals is the devices are in their infancy in terms of deployment. Many people have few or no devices in their homes, however these devices are starting to gain popularity. As these devices become more common there will be more attacks against them. The third reason that ransomware is a less popular method of infiltrating is the information on these devices is less likely important or valuable to the user and therefore its less likely that the user will pay a ransom to regain access to the device. As well as if the user of a given device has the information backup separately then it is unlikely that they would pay for access to that device.

2.2 Cloud Storage in IoT

Cloud storage is common on many devices that are commonly used in people's everyday lives and IoT devices are no exception to that. The usage of IoT devices in people's everyday life has increase as these devices become increasingly available. These devices are managed by cloud services. The devices operate with the manufactures cloud services (Philips, Sylvania, etc. as well as cloud provider services (Google) [1]. The devices not only communicate with their respective providers clouds they also communicate with the cloud providers. The typical process of connecting the devices to these clouds is the user connects the device to the device providers cloud using the provider's phone app. Then using the third-party cloud provider's interface connect the providers cloud to the third-party cloud. When the user accesses their devices through the third-party's interface cloud authentication ensures that the user has authority to control these devices [1].

3. METHODOLOGY

3.1 Environment – Devices

For this experiment the IoT devices had to be set up using their providers apps and then connected to the google home app to be controlled by the google nest mini. Table 1 shows the name of each device as well as what kind of device each device is, the app that was used to connect the device and the connection the device used.

Table 1. Devices and Their Interfaces

Device Name	Device Type	App Name	Connection Method
Google Nest Mini	Smart Speaker	Google Home App	WiFi
Philips Hue bulb	Bulb	Google Home App	Zigbee
Feit bulb	Bulb	Feit Electric App	WiFi
Sylvania Bulb	Bulb	Sylvania App	WiFi
GE Bulb	Bulb	Google Home App	Zigbee
Merkury Bulb	Bulb	Geeni App	WiFi
Merkury Smart Wi-Fi camera	Camera	Geeni App	WiFi
Kasa Smart Plug	Plug	Kasa App	WiFi
Kasa Spot smart camera	Camera	Kasa App	WiFi
Google Home Smart Camera	Camera	Google Home App	WiFi

3.2 Environment – Platform and Tools

Platform - The platform used for this project was Kali Linux. This was chosen because it is a Linux machine used for penetration testing.

Tools

- Wireshark – this was used to sniff the packets from the devices.
- Nmap – this is a built-in command to find devices that are operating on a network.
- Network Miner – this was used to search the packets for important information.
- Airmon-ng – was used to set up a man in the middle attack by managing network processes.
- Airodump-ng – was used to set up a man in the middle attack by using this command to change the status of wlan0 to monitor.

3.3 Methods

The first technique was to connect the Sylvania, Feit, Philips's hue, and GE bulbs to the google nest mini. Then, track the packets over Wireshark in search of weaknesses or helpful information. Then use the information to infiltrate the device with ransomware. After

infiltrating the bulb use the SDK to confirm the ransomware was on the cloud. Finally, send the ransomware through the cloud to the google nest mini and lock that device. Unfortunately, the devices chosen originally sent information through the Zigbee protocol and required a cost prohibitive Zigbee sniffer.

The Merkury Bulb, Merkury Smart WiFi camera, Kasa Smart Plug, Kasa Spot smart camera, Google Nest cam Indoor camera all operated under the wifi protocol meaning they could be sniffed by Wireshark without any additional devices. These were the new devices chosen for this experiment. The question then became could these devices hold useful information in their packets.

The first step was connecting all the devices to their respective apps and the google nest mini. Each device had its own app that is needed to be connected to and then connected to the google home app. After they were connected to the google home app, they could then be controlled by the Google Nest Mini. From there the technique was to use Wireshark to analyze the packet communication.

Wireshark works by reading the packets the are sent and received from each IP address on the network. This meant that the Ip addresses of each IoT device needed to be found.

The process of finding the IP addresses of all the devices was:

1. Nmap to find all active devices on the network.
2. Ipconfig on the Kali Linux to cross off that IP from the list.
3. Do the same for all devices on the network that did not pertain to this experiment.
4. Go to settings to find the IP of android phone that the apps were downloaded on.
5. Discover the Merkury Bulb, Merkury camera, and Google nest in their respective apps.
6. Ping remaining IP addresses found by Nmap and turning a device off and seeing if the ping stops to discover the IPs.

The next step was finding the packet of each device using their IP addresses. Originally the only packets Wireshark was interpreting was broadcast UDP packets, and the packets sent to and from the Kali box. These packets do not provide any information that is helpful for this experiment. So, to overcome this issue a man in the middle attack was necessary.

A man in the middle, “MiTM”, attack is when there is an attack that intercepts the packets before they reach the router and enables the viewer to see what type of packets are being received by the router.

The “MiTM” attack was created in kali Linux by:

1. In the terminal type iwconfig and find the wireless interface name, in this case it was wlan0.
2. Verify the Wi-Fi adaptor is capable for monitoring mode. Type iw list to learn the capabilities of the adaptor.
3. Kill interfering processes using airmon-ng
4. Using airmon-ng create a monitor mode interface.
5. Ensure the Wi-Fi adaptor is operating in monitor mode by using airodump-ng.
6. Open Wireshark and sniff the interface that was created.

The next problem was the packets were still encrypted; this means that the only protocol viewed was 802.11. This is still unhelpful and is not very insightful. The solution to this problem was to save the Wi-Fi password in Wireshark. The Wi-Fi network used was a personal network where the password was known. Discovering the password through surveillance is technically illegal in the US. After saving the Wi-Fi password the packets’ protocols were able to be viewed. The specific packet protocol that contains important information is TCP. The packets that were pulled from Wireshark and saved to be read were large TCP files that were the most likely to hold important information.

Unfortunately, Network Miner, which is a data carving tool used to find sensitive information and files of certain specified types, was unable to find helpful information that would lead to gaining access to the IoT devices. Network Miner is also a file carving tool and was unable to find any files in the packets.

4. FUTURE WORK

One barrier to success was the Zigbee protocol. Future research could include how to infiltrate Zigbee devices. Comparing the security of Zigbee devices versus Wi-Fi enabled devices. Some devices require a hub to convert Zigbee to Wi-Fi, research could be done on the security of the hub.

There are many tools the penetration testers as well as malicious actors use to access sensitive information. Future Research could be done on the various tool and access the ones that are most effective. There are also many other ways of infiltrating a device. An API could be used to send malicious information to the device or cloud.

5. CONCLUSION

In conclusion the hypothesis of authentication and communication mechanisms of cloud Storage from IOT devices are vulnerable to ransomware was unable to be confirmed within this paper. This paper talked about the various types of IoT devices and the analyzed the ones that were the most efficient for the scope of this paper. The security of these devices was scrutinized in various ways and the outcomes were unsuccessful in infiltrating these devices. This paper is unable to prove that IoT devices are weak to malicious actor’s techniques however, that does not prove that these devices are secure against all infiltration techniques or malicious actors.

6. ACKNOWLEDGMENTS

Special thanks to Eric Wright for advising on this project.

7. REFERENCES

[1] Bin Yuan, Yan Jia, Luyi Xing, Dongfang Zhao, XiaoFeng Wang, & Yuqing Zhang (2020). Shattered Chain of Trust:

Understanding Security Risks in Cross-Cloud IoT Access Delegation. In 29th USENIX Security Symposium (USENIX Security 20) (pp. 1183–1200). USENIX Association.ACM SIG PROCEEDINGS template. http://www.acm.org/sigs/pubs/proceed/template.html.

[2] Calvin Brierley, Jamie Pont, Budi Arief, David J. Barnes, and Julio Hernandez-Castro. 2020. PaperW8: an IoT bricking ransomware proof of concept. In Proceedings of the 15th International Conference on Availability, Reliability and Security (ARES '20). Association for Computing Machinery, New York, NY, USA, Article 82, 1–10. DOI:https://doi-org.wsuproxy.mnpals.net/10.1145/3407023.3407044Mackay, W.E. Ethics, lies and videotape... in Proceedings of CHI '95 (Denver CO, May 1995), ACM Press, 138-145.

[3] Jianliang Wu, Yuhong Nan, Vireshwar Kumar, Mathias Payer, & Dongyan Xu (2020). BlueShield: Detecting Spoofing Attacks in Bluetooth Low Energy Networks. In 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020) (pp. 397–411). USENIX Association.

[4] Muna Al-Hawawreh and Elena Sitnikova. 2019. Industrial Internet of Things Based Ransomware Detection using Stacked Variational Neural Network. In Proceedings of the 3rd International Conference on Big Data and Internet of Things (BDIOT 2019). Association for Computing Machinery, New York, NY, USA, 126–130. DOI:https://doi-org.wsuproxy.mnpals.net/10.1145/3361758.3361763

[5] Pranshu Bajpai, Richard Enbody, and Betty H.C. Cheng. 2020. Ransomware Targeting Automobiles. In Proceedings of the Second ACM Workshop on Automotive and Aerial Vehicle Security (AutoSec '20). Association for Computing Machinery, New York, NY, USA, 23–29. DOI:https://doi-org.wsuproxy.mnpals.net/10.1145/3375706.3380558

[6] Zhen Li and Qi Liao. 2020. Ransomware 2.0: to sell, or not to sell a game-theoretical model of data-selling Ransomware. In Proceedings of the 15th International Conference on Availability, Reliability and Security (ARES '20). Association for Computing Machinery, New York, NY, USA, Article 59, 1–9. DOI:https://doi-org.wsuproxy.mnpals.net/10.1145/3407023.340919

“Looks Like Its Rush Hour Again” – How Botted Users Have Increased The Traffic Of Websites.

Alireza Shahrokhi, Author
Winona State University
175 W Mark ST
Winona, MN, 55987
Alireza.Shahrokhi19@gmail.com

Abstract
The main purpose of this paper is to discuss the ideas about how botted users are affecting websites. The problem that comes when bots take over websites is that these botted users cause so much traffic that websites cannot scale well enough. Websites after the holidays see a mass decrease in the number of users on their sites. However, in 2021 and with years to come this has changed where we will see an increase. It is estimated that the number of bots on a website through 2021 is about 35% of all traffic. In this paper I will discuss that we have increased this number and will see a sharp increase with years to come.

Keywords
Automated Users; Website Traffic; Website Scaling; Bots

1. Introduction
This study will heavily focus on the effects of botting and how we have seen an increased number of botted users in 2021 and will most likely see an increase of these users in years to come. The reason for the increase of botted users in 2021 is a question that has many people wondering why. With COVID-19 we have seen mass numbers of people come and start shopping in an online experience for the first time, or at least the user base that shops online have increased tenfold. Retailers’ websites were ready for the huge masses of people that have changed to online shopping rather than in store but were not ready for the mass amounts of increase when it comes to botted users. In this study we will dive deeper into how this will affect the user experience and how security can also be affected by these botted users. Approaches that I will take to show the importance of this increase I will be looking at live traffic of websites during regular business and during drops to show the increase of these users as well as show how even after the holidays.

Where we have seen traffic data numbers that usually has a sharp decrease in traffic increase heavily in February 2021. There are several methods in the past that retailers have taken to get better ideas on who the botted users are. However, the most used method is the naming convention each browser that we use such as chrome has a name attached. Firewalls will see these names and flag browsers that are random string of letters and numbers. It is not as easy as just looking at your firewall and seeing an IP that you “think” might be a botted user who may be a customer just trying to use their site. This issue is huge because the retailers may ban this IP for someone who has not done anything wrong.

This paper will discuss issues with API (Application Programming Interface, information being pulled from source) requests per second when these botted users are active. This can cause severe issue with security and how user information can be leaked during times of high user activity. Part of this paper will be read on a criticism with how theses botted users are handled. It is very important to understand that my criticisms lie only with how these botted users can checkout multiple products. My research project is important because we live in a world that is changing every day. People are out there that use online shopping every single day and with Covid this number is higher than ever before.

Proceedings of the 21st Winona Computer Science Undergraduate Research Seminar, April 19, 2021, Winona, MN, US.

Figure 1. Monthly Walmart traffic data from May 2019 – September 2020. Each point represents millions of users that have entered the site. During holiday times we see an increase in number of users

1.1 What is website traffic?

Website's traffic is a very simple way of seeing how many users have entered your website. Each person that joins your website should always be considered one person. With website traffic you can also have a much better understanding of your metrics. Who purchased something when they came to your website, how long did they stay on your given page? You can imply these into multiple different categories, such as how well a sale did and how many people increase did your site receive. All these are important topics however in our case this helps significantly with finding out the number of bots or estimated number of bots on a website. Remember that each time you open a website on a browser that would mean a new person is equated to the current number of users on the given firewall.

2. Hypotheses/Questions

My hypothesis is based on the increase of botted users on retail websites. "Botted user on retail websites have increased by 20% over the last year" The people using bots have increased substantially over the past year. Does this have to do with the global number of users coming to online shopping or just increased tasked users. This question is important as it sets up our entire project going forward.

3. Methods

3.1 Mr. Bearden Interview

During a conducted interview with Dan Bearden from BestBuy. Mr. Bearden stated "Botted users are taking over websites as we know it. People buying mass products to sell for a much higher cost. These bots have negative effects on BestBuy.com and we try everything in our power to stop them." Dan reported that these bots run hundreds or in some cases thousands of tasks which can equate to 2000 people each or even more. This is important because that would mean for each task that is ran on BestBuy.com that each task would equate to one person using that given site. During BestBuy.com drops for the new PlayStation a shock drop was announced on the company twitter (https://twitter.com/BestBuy). Once the time was set for the drop. Dan illustrated that BestBuy went from having a couple thousand requests per minute to having hundreds of thousands of requests per second. This caused major security issue as these many requests caused one very important API to become public. This API oversaw all customer purchase 4-part keys. These keys could allow anybody who has the customers 4-part- to change important information regarding the order they had placed. This important information could include changing shipping address, customer name, quantity, and other important details regarding the receipt.

3.2 Walmart Traffic Data

To better understand how bots are equating to large amounts or traffic I took a deeper look into the traffic data of Walmart over the past year. When we take a closer look, we see that in May of 2019 there were 332 million users that used Walmart.com. However, when we look at the same month in 2020, we see that this number has increased to 438.5 million users that's difference of 106.5 million users. This is almost a 25% increase of users in just one year this data can be found in Figure 1 (Sabanoglu,2020). Now you may be asking yourself how this is not just because of a global

increase of users during COVID-19. When we look at all the traffic data with other months, we can see that every other month still had a massive increase in the number of users per month. Moving from May to June there is a sharp decrease in the number of users. This is strange because even after a sharp decrease in amount of users Walmart still had seen the most users ever prior to 2020. Now let us compare Walmart's January 2020 trends to the current 90-day trend. In figure 2 even after the holidays where there have been sharp declines in the past that Walmart's traffic is growing every day (Alexa, "Alexa Rank 90 Day Trend", 2021). This growing traffic is based on the exclusive drops on Walmart's website over the last month. Botted users increase the traffic of websites and the graphs show that this is the case. Over the next coming months, I predict that this number will grow even higher.

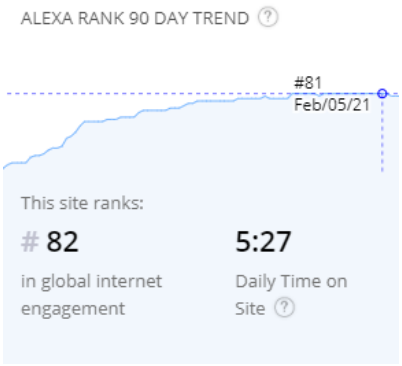

Figure 2. Alexa website traffic data Walmart February 2021. Showing increase in traffic data through February. This differs from 2019 where we see a sharp decrease in website traffic.

Figure 3. Alexa website traffic data BestBuy March 2021. Showing decrease in the traffic on BestBuy.com. This trend plateaus however starts to decrease shortly after.

Figure 4. Alexa Website traffic data Target April 2021. The 90-day trend continues its path of decreasing traffic. We now can see information for each side of the retail market after.

4. Results

4.1 BestBuy Traffic

As we can see my hypothesis has been disproven from a statistical aspect. Figure 3 and Figure 4 show a decrease in trend of website traffic over the last 90 days day (Alexa, "Alexa Rank 90 Day Trend", 2021). Figure 3 represents website traffic data we have analyzed for BestBuy.com. During this 90-day period of research we have seen Best Buy traffic be stagnant and quite volatile at the same time. Although my hypothesis got disproven the data and information that we have shown, traffic can change very quickly and show us different trends. This result shows us the power of bots. With thousands of tasks running bots can show sharp decrease or increase in traffic. (Alexa, "Alexa Rank 90 Day Trend", 2021).

4.2 Target Traffic

Target.com shows us similar data, in a much less volatile sense. While BestBuy.com had very sharp decreases in traffic Target showed this same downward trend with a much more linear trend. However, when we take a deeper look into when we see drops in traffic this data tells us a different story. With many increases to bot protection automated users (bots) can no longer monitor these websites 24/7 like they have in the past. With Best Buy there is a clear point where traffic data decreased dramatically. However, it is seen that this curve flattens very quickly. This is due to hyped releases that automated users can take advantage of and buy products in bulk.

4.3 Retail Market Traffic

Figure 5. Shows the total amount of users that accessed the website in that given month. The trends are almost identical to each other in traffic increase and decrease. April 2021 approximates user traffic as the month is not completely over yet. All that was done was taking current month from April 1 – 15 and doubling to approximate the end of the month. As GPU stock increased at the end of march at this point is where we see sharp increase in traffic. Showing that botted users are controlling the traffic of retail websites.

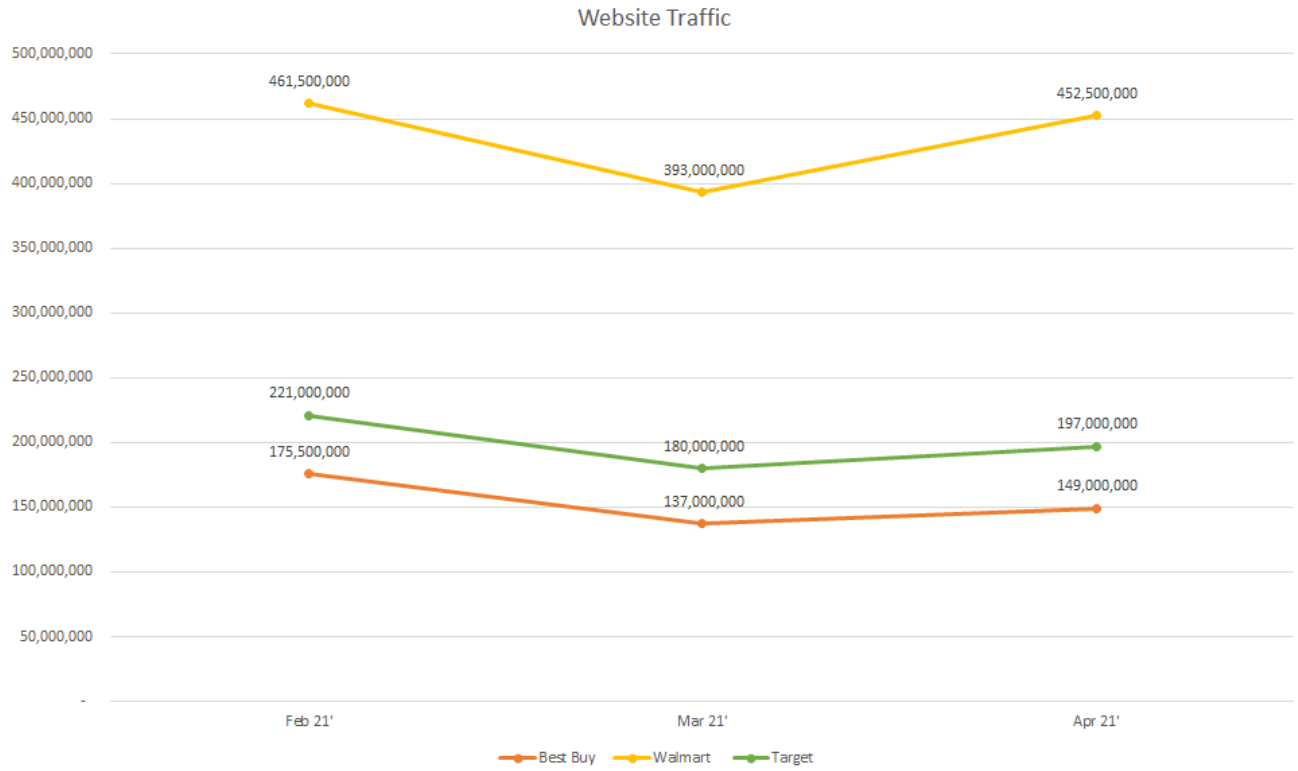

Figure 5. Graph representing total amount of user traffic per month on each respective website. This graph is used to show the linear increase and decrease in retail website traffic.

4.4 How can we stop bots?

There are many ways that bot security is trying to upgrade to prevent the number of botted users on their site. Many are flagging Ips with "Robots" on them but a lot of time these users are real customers. When your IP gets flagged for having a robot this could mean many different products. Bot protection have flagged Ips for having even rumbas on their network. The best possible way in my opinion to prevent botted users is multifactor authentications. Best Buy does this very well, they require you to have an account on their website to checkout any products. This would be considered 1 step of factor authentication. However, you can add to this. Once you are randomly selected in the queue you should have to input a code that has been sent to your email. Now this is multifactor authentication, to take this to another level we can require that people enter the code received from their email and to input a code that was sent to their mobile device. This will severely limit the number of botted users. With 3 step authentication botted users will not have enough time to checkout multiple products as one transaction can take up a lot of time. Many botted users also do not have multiple phone numbers to checkout another time.

5. Conclusions

This paper describes how botted users have been tormenting retail websites over the past year, and important evidence that proves these findings. Botted users can cause major security risks and can severely downgrade the performance of the website. The ways we can prevent bots from taking over and to try to severely limit the number of bots that allow to checkout on any given website. In short what I suggest is to implement more authentication before checkout. People should have to put input a code into the checkout screen before they can proceed with placing the order.

6. Acknowledgments

May 2021 I will be a front-end developer for BestBuy and bot security of a website is one of our biggest worries. Over the semester I have learned about botting and the effects it has. I will continue my research even after this course is done. Thank you to all the developers at BestBuy that assisted me and gave me the knowledge I needed. Thank you to my professors Dr.Iyengar and Dr.Zhang for assisting me in every step within this process.

7. **References**

[1] Shahrokhi, Ali, and Dan Bearden. “Botting within BestBuy.” 2 Feb. 2021.

[2] Statista. (2020, December 1). *Total global visitor traffic to Walmart.com 2020*. https://www.statista.com/statistics/714568/web-visits-to-walmartcom/

[3] *Alexa - Competitive Analysis, Marketing Mix, and Website Traffic*. (2021, March 31). Alexa. https://www.alexa.com/siteinfo