

Cybersecurity Team Capstone Report

ADVISOR: DR. CAVALCANTI

ANTHONY QUINTERO, JESSE VILLANUEVA, HAYDEN FLAGG, PIERCE
PETERSON

Contents

Introduction.....	3
Tools and Applications	3
Communication.....	3
Hosting Service.....	4
Website	4
Progress.....	5
Sad Server	5
Login Page	5
Sign Up Page.....	6
Index Page.....	7
Account Info/Grocery List/Inventory Pages	8
Exploits	10
Solutions to exploits.....	10
SQL Extension	11
Dropdown Injection	11
Text field Injection.....	11
Login password attack	12
Account Info UserID Spoof	12
Account Info Table Change - Users.....	12
Grocery List Dump	12
Inventory Multiple Categories	12
Testing.....	13
Results.....	13
Good Server – Firebase.....	13
Languages	14
Why is it more secure?.....	14
SQL injection	14
DDoS.....	15
Port Access.....	15
Password hashing.....	16
Future	16

Appendix.....	17
Entity Relationship Diagram.....	17
User Manual.....	18

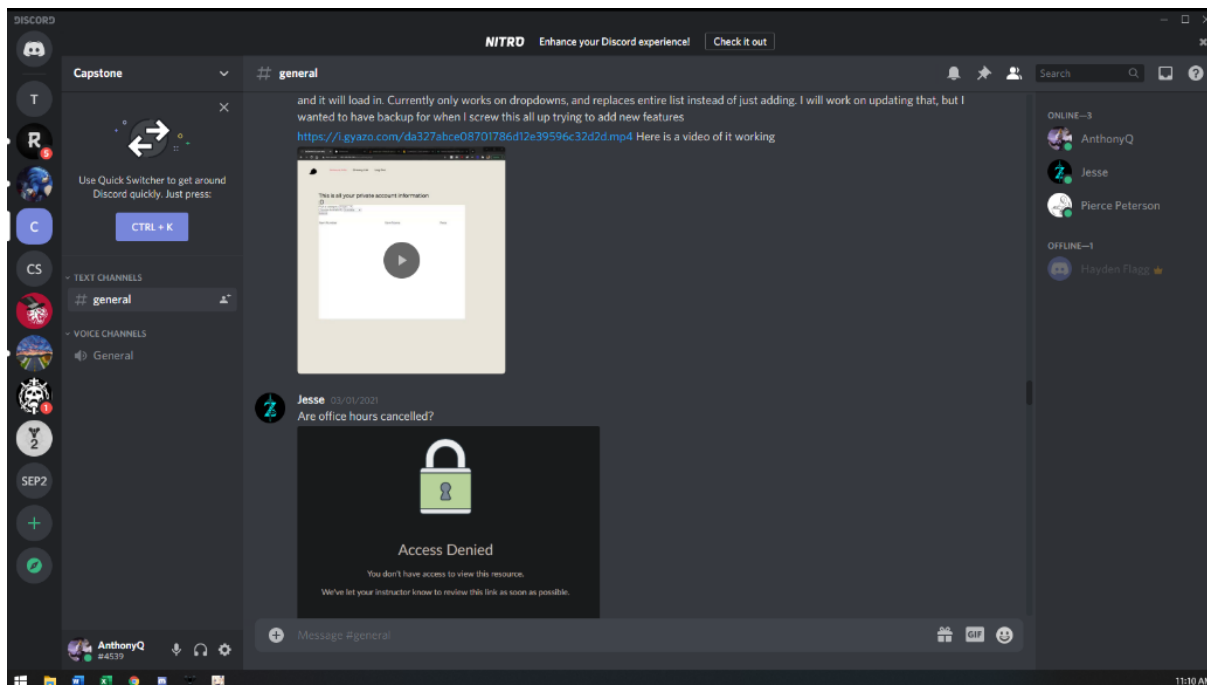
Introduction

For our project we chose the subject of cybersecurity. We planned to make two websites/servers. One server, the Sad Server, is meant to be vulnerable to cyber-attacks allowing us to hack it using methods like SQL Injection and DDOS attacks. The second server, the Good Server, is supposed to be an improvement on the Sad Server. It is meant to be safe, secure, and impenetrable. Our teacher suggested using Black Arch a penetration testing application that provides cybersecurity tools. Although we could not get it to hack our Sad Server, we moved on to developing our own SQL injection application.

Tools and Applications

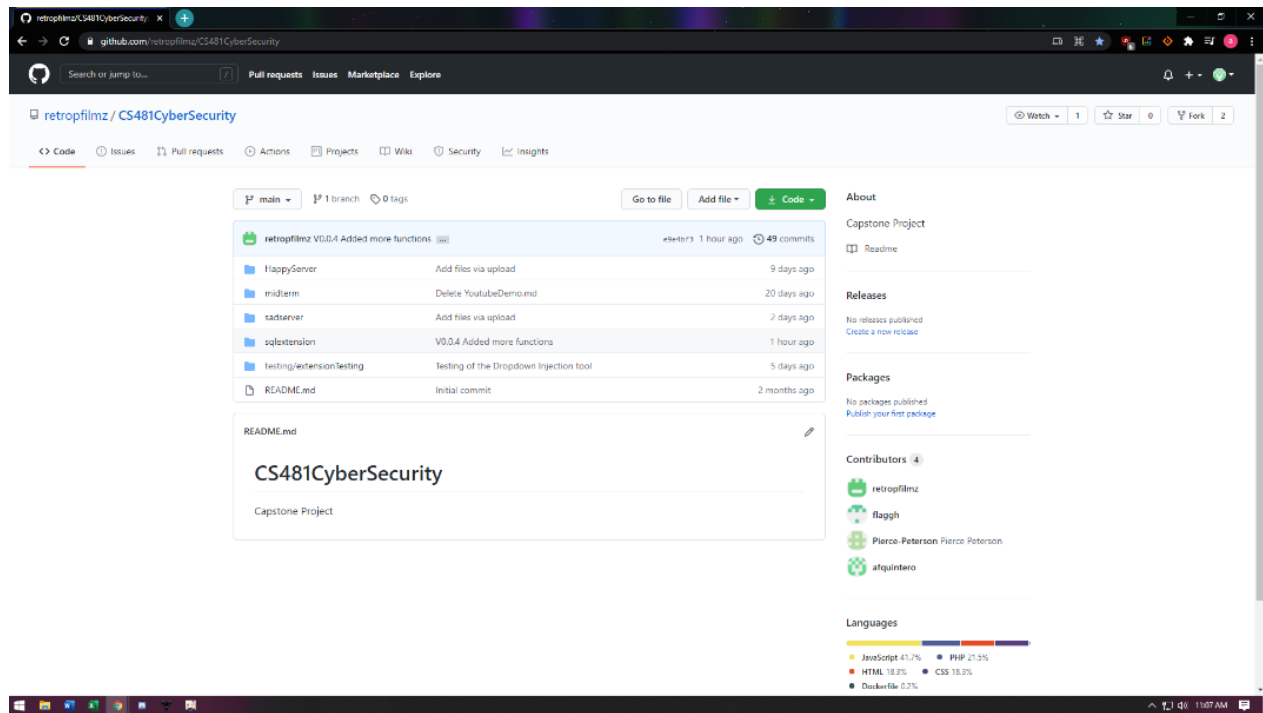
Communication

Our group communicated through Discord, a voice, video, and text communication application. We had 2 or 3 meetings a week and sent texts when a meeting was not needed. We could also share files, but it wasn't our main method of exchanging files.



Hosting Service

We used Git hub to share and access our files. Git Hub is a web-based platform that allows software developers to collaborate and control versions of their software.



Website

We decided to go with a shop type of website where you can view available products and purchase them. We implemented a login and registration system as well. To host our websites, we used a Docker container running LAMP and Firebase for the Sad Server and Good Server respectively. Docker is a virtualization platform for developing applications that use containers. Developed by Google, Firebase is a platform for developing mobile and web applications.

Progress

The two websites work as intended so far. They allow users to both login and signup. There are 2 pages, account info and grocery list that allow you to view products available. Dr. Cavalcanti suggested we use Black Arch for hacking our website. While it couldn't hack our good server like intended it also couldn't hack our sad server. We tried SQL Injection and kept getting no query string as an error.

```
jec [19:32:34,561] Undefined URL protocol, forcing to [http://]
a M [19:32:34,564] Starting new injection: http://capstone.retrop.net/accountinfo.php
00 [19:32:34,564] No query string
00 [21:50:14,478] Starting new injection: https://store-16edf.firebaseio.com
00 [21:50:14,478] No query string
```

Although we found another way to alter our website. Our sad server lists products that are available for sale and we were able to change whether the items were available or not through editing the URL of the accounts page. We've also developed a web-based extension that manipulates the queries by changing various parts of the page.

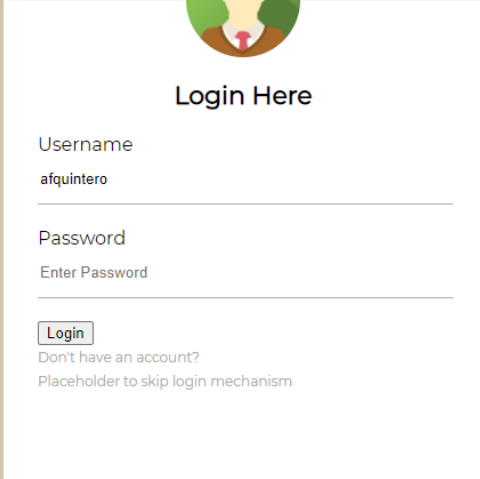
Sad Server

Our Sad Server is locally hosted at <http://capstone.retrop.net/>. It uses PHP, HTML, CSS, JavaScript, and SQL. HTML and CSS is used for the front end. PHP is used for the login and signup features to access our database and SQL is what we used for our database. This sad server is meant to be vulnerable to hacking like SQL injections and DDOS attacks.

Login Page

The login page is the first page new visitors will see. It allows previous users to login and new users to access the sign-up page. It asks the user for their username and password. For ease of


use we've also included a skip login link to go to the account information page. Using this login bypass will not allow the account page to work to full capacity, but will allow you to access the rest of the pages. If you needed to make an account, you can click on "Don't have an account?" which will bring you to the sign-up page.



The image shows a login form centered on a light brown background. At the top of the form is a circular icon of a person with dark hair and a red tie. Below the icon is the title "Login Here". The form contains two input fields: "Username" with the text "afquintero" and "Password" with the placeholder text "Enter Password". Below these fields is a "Login" button. At the bottom of the form, there are two links: "Don't have an account?" and "Placeholder to skip login mechanism".

Sign Up Page

The sign-up page is like our login page, it asks for your username, password, and email. You will also need to confirm your password. The website will make sure your password matches with the confirm password text field. Once you click register it will save your account details to our database. If you already have an account, you can click "sign in" at the bottom to go back to our login page.



Sign-up Here

<
Username

Email

Password

Confirm password

[Register](#)

Already a member? [Sign in](#)

Index Page

Once you login you will be taken to our Index/Home Page. If you sign up, you will be automatically logged in and brought to this page as well. This page will let you know that you successfully logged in. On this page you have you have the option of logging out or accessing our account info page. If you had logged in recently, the website will remember and you will be taken straight to the index page skipping the login and sign up pages entirely.

Home Page

You are now logged in

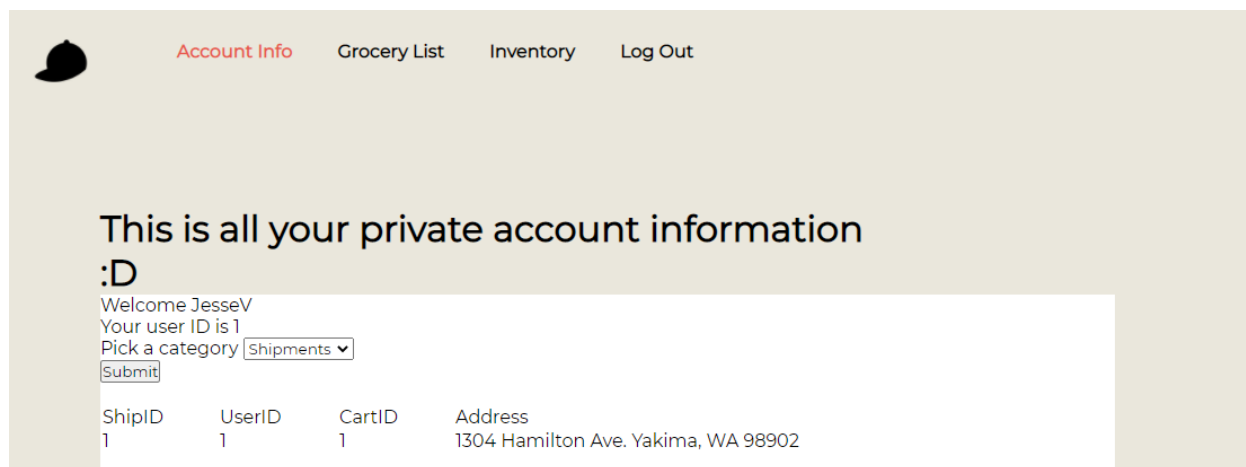
Welcome **afquintero**

[logout](#)

[Account Info](#)

Account Info/Grocery List/Inventory Pages

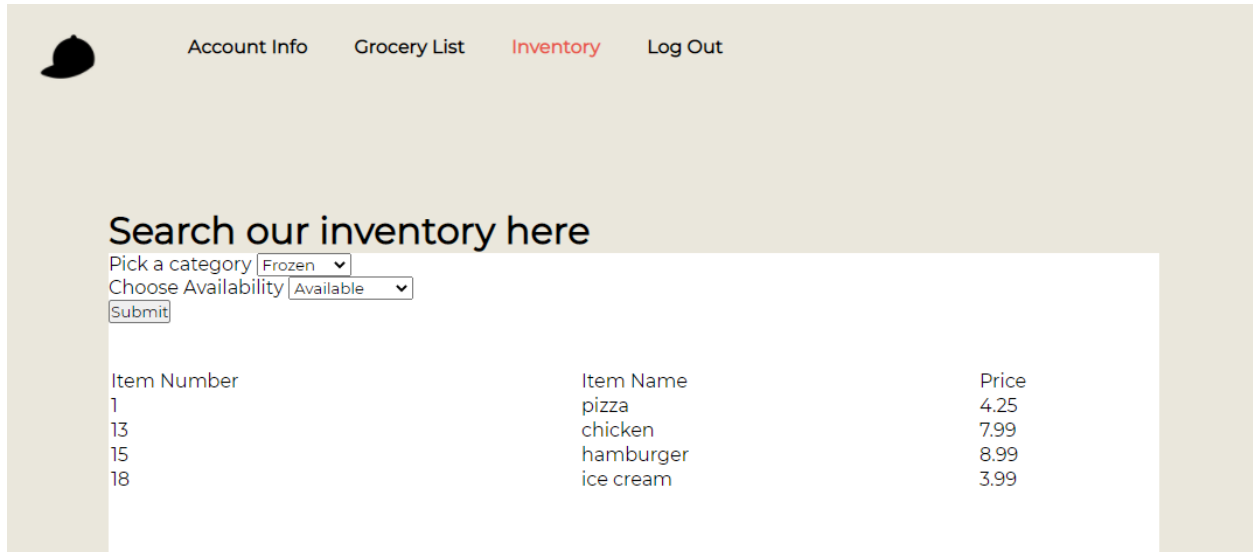
Our Account Info page shows you information related to your account. The dropdown lets you swap information based on your account number which was automatically generated when the account was created. There are two options in the account info page to access the tables in our database. The query checks your selected table for items related to your user ID. In our Sad Server this this can be problematic because if you dig enough into the website you can edit your user ID locally and inject into the query allowing you to see other's information. This can be combined with injecting another object into the dropdown list to view other tables that also contain user ID. In our case since our shipping and shopping cart is related to the user ID, the users table also has this tag.



The screenshot shows a web application interface with a navigation bar at the top. The navigation bar includes a logo on the left and four links: 'Account Info' (highlighted in red), 'Grocery List', 'Inventory', and 'Log Out'. Below the navigation bar, the main content area has a heading 'This is all your private account information :D'. Underneath the heading, there is a form with the following text: 'Welcome JesseV', 'Your user ID is 1', 'Pick a category' followed by a dropdown menu showing 'Shipments', and a 'Submit' button. Below the form, there is a table with four columns: 'ShipID', 'UserID', 'CartID', and 'Address'. The table contains one row of data: '1', '1', '1', and '1304 Hamilton Ave. Yakima, WA 98902'.

ShipID	UserID	CartID	Address
1	1	1	1304 Hamilton Ave. Yakima, WA 98902

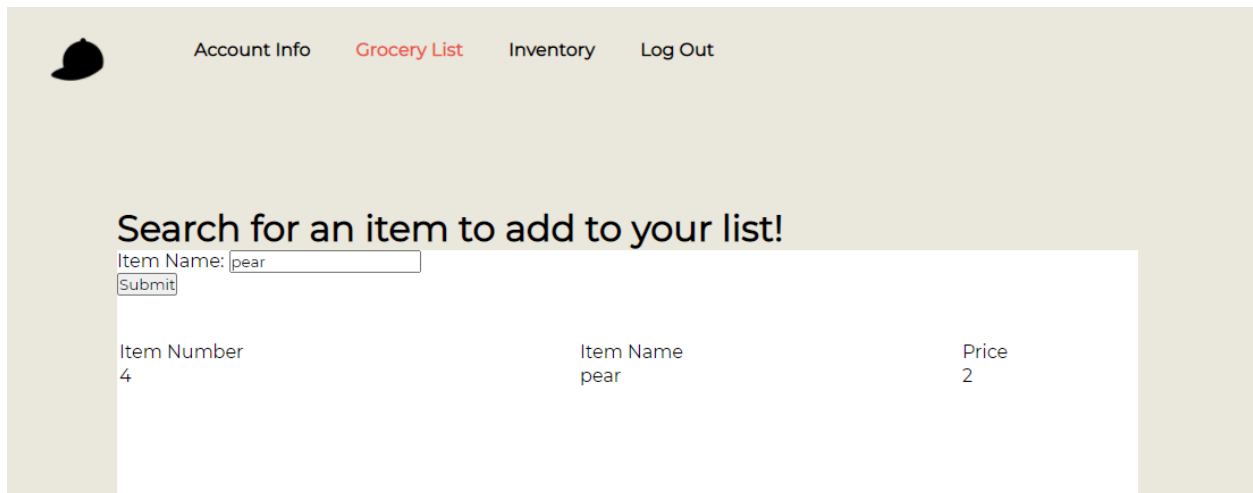
On our inventory page, you can view our products by utilizing the two drop down windows. The first drop down window has choice between produce and frozen, whereas the second drop down window allows you to view either the available or unavailable products.



The screenshot shows the 'Inventory' page of a web application. At the top, there is a navigation bar with a profile icon, 'Account Info', 'Grocery List', 'Inventory' (highlighted in red), and 'Log Out'. Below the navigation bar, the heading 'Search our inventory here' is displayed. Underneath the heading, there are two dropdown menus: 'Pick a category' with 'Frozen' selected and 'Choose Availability' with 'Available' selected. A 'Submit' button is located below these dropdowns. The search results are displayed in a table with three columns: 'Item Number', 'Item Name', and 'Price'.

Item Number	Item Name	Price
1	pizza	4.25
13	chicken	7.99
15	hamburger	8.99
18	ice cream	3.99

The grocery list also allows you to view our products albeit through a search bar instead. You just enter your search term like “pear” and push the submit button. Just like the inventory page you will be given its’ item number, name, and price.



The screenshot shows the 'Grocery List' page of a web application. At the top, there is a navigation bar with a profile icon, 'Account Info', 'Grocery List' (highlighted in red), 'Inventory', and 'Log Out'. Below the navigation bar, the heading 'Search for an item to add to your list!' is displayed. Underneath the heading, there is a search bar with the text 'Item Name: pear' and a 'Submit' button. The search results are displayed in a table with three columns: 'Item Number', 'Item Name', and 'Price'.

Item Number	Item Name	Price
4	pear	2

Exploits

The Sad Server uses the URI or Uniform Resource Identifier for the form submission. URI is a sequence of characters that are used to identify resources used by web technologies. Other values can be added to the URI which may be different from what is normally allowed. This SQL server uses an availability column for unreleased, available, and out of stock products. Although our availability setting on the page only allows Available or Out of Stock. Manually setting the availability to 0 allows user to view items they should not be able to see.

```
mysql> Select * from Items
-> ;
```

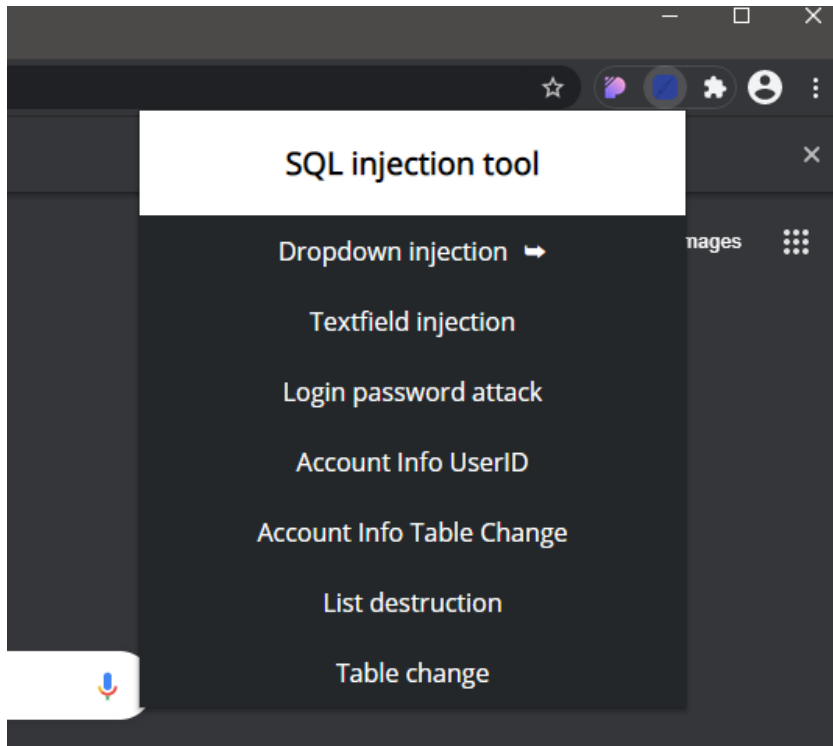
id	name	price	category	availability
1	pizza	3	frozen	1
2	ice cream	3	frozen	2
3	oranges	3	produce	0
4	pear	2	produce	1
5	apple	3	produce	1
6	banana	2	produce	2
7	watermellon	6	produce	0

Solutions to exploits

To avoid these exploits, you can avoid URI for queries but using post instead of get. Taking the category from the fields directly can help prevent unwanted inputs or using a list can prevent unwanted inputs without use of console commands. More options available are filtering the inputs to make sure there are no special characters that would allow the query to be manipulated and using more secure queries.

SQL Extension

We developed an injection tool you can install into your preferred chromium-based browser. The injection tool has two generic methods that can be attempted on any page, and 5 that were made specifically for our websites.



Dropdown Injection

Generic method to target all dropdown menus and replace entries with the option that allows for SQL injection. Can be used on the Inventory page to show contents of all categories in Item table.

Text field Injection

Generic method to target all text fields and insert values for SQL injection. Can be used on the Login or Grocery List page to get all items.

Login password attack

Targets the password input field and inserts query injection. To be used on the login.php file of the Sad Server. When logging in after using this, all usernames and emails will be listed on the redirect to index.php.

Account Info UserID Spoof

Targets the user ID number on the Account Info page. Since the query references the user's ID, we can inject to get more than just our specified items. When submitting a query after running this command you see info for all user IDs.

Account Info Table Change - Users

Targets the dropdown menu on the Account Info page. Adding an option to the dropdown allows us to view tables we are not supposed to view. After executing this the Users table becomes part of the dropdown menu. Using this alone will show you all of your account information. Using this in combination with Account Info UserID Spoof will show you all user data.

Grocery List Dump

Targets the search text field on the Grocery List page. This query uses the input to find items that match. Submitting a query after running this command will dump the entire list of items.

Inventory Multiple Categories

Targets the dropdown on the Inventory page. Allows for a feature that should really be implemented by a real website but could possibly be used maliciously. This example adds the option for both Frozen and Produce to be queried at the same time.

Testing

Of course, we also tested each method with all passing and performing what they needed to. We tested this on both Google Chrome and Microsoft Edge. Testing of all JavaScript methods were done using Jest within VSCode. Jest was used as it allowed us to test the main feature of the extension, DOM manipulation. Each test generates the respective tag and then runs the command. After the command is ran, it checks to make sure the injection into the DOM works.

```

PASS ./content.test.js
  ✓ Test dropdown injection of 1=1 (13 ms)
  ✓ Test dropdown injection of 1=1 with no dropdown (1 ms)
  ✓ Test text field injection of 1=1 (2 ms)
  ✓ Test injection into password text field
  ✓ Test injection into user id span (1 ms)
  ✓ Test manipulation of dropdown to add user table option (2 ms)
  ✓ Test injection into search text field
  ✓ Test manipulation of dropdown to add a combination option (2 ms)

-----|-----|-----|-----|-----|-----
File      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----
All files |    100   |    100   |    100   |    100   |
  content.js |    100   |    100   |    100   |    100   |
-----|-----|-----|-----|-----|
Test Suites: 1 passed, 1 total
Tests:       8 passed, 8 total
Snapshots:   0 total
Time:        2.345 s
Ran all test suites.
PS E:\Documents\CS481\testing\extensionTesting>

```

Results

Good Server – Firebase

For our Good server we used a third-party source called Firebase to emulate our good server.

Like our Sad Server, the Good Server contains our databases full of user information and items or products in our store. Our Good Server can be found at <https://store-edf16.firebaseio.com>.


Firebase provides an advantage that it's protected against cyber-attacks that our Sad Server may is not protected against.

Languages

HTML & CSS also make up the body of the Good Server. JavaScript retrieves data from our server and displays the needed data onto the website. The JavaScript interacts with our database through Firebase.

Why is it more secure?

SQL injection


Account Info
Grocery List
Inventory
Log Out

Search our inventory here

Pick a category Frozen
Choose Availability Available

Item Number	Item Name	Price
1	pizza	4.25
2	ice cream	3
3	oranges	4
4	pear	2
5	apple	0.99
6	banana	3
7	watermellon	6
8	eggs	2
9	lays chips	4
10	oreos	2.99
12	chips ahoy	2.99
13	chicken	7.99
14	milk	2.99
15	hamburger	8.99
16	lemonade	2.99
17	lasagna	8.99
18	ice cream	3.99
19	lettuce	2.99
20	fireball	13.99
21	olive	1.99
22	flour	5.99
23	ramen	0.99
24	cashews	3.99
26	cereal	4
27	kit-kat	1.5
28	grapes	6
29	bread	2
30	cheese	2

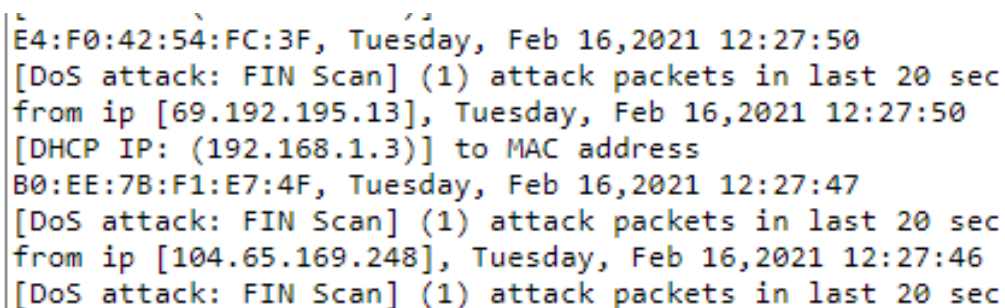
Firebase doesn't use SQL or PHP to create and manipulate its database, rendering SQL injection useless.

DDoS

DDoS or Distributed Denial of Service attacks takes advantage of the limitations of websites. The website is flooded with multiple systems to attack the bandwidth of the site.

Firebase is run by another more professional party with more protections and regulations on what goes through the database and can handle more simultaneous inputs than our other server can.

Below is a screenshot from when our server was attacked, many connections took down the home internet of the host. After the attack ended they port scanning was attempted but thankfully only the necessary ports were open and were contained.



```
E4:F0:42:54:FC:3F, Tuesday, Feb 16, 2021 12:27:50  
[DoS attack: FIN Scan] (1) attack packets in last 20 sec  
from ip [69.192.195.13], Tuesday, Feb 16, 2021 12:27:50  
[DHCP IP: (192.168.1.3)] to MAC address  
B0:EE:7B:F1:E7:4F, Tuesday, Feb 16, 2021 12:27:47  
[DoS attack: FIN Scan] (1) attack packets in last 20 sec  
from ip [104.65.169.248], Tuesday, Feb 16, 2021 12:27:46  
[DoS attack: FIN Scan] (1) attack packets in last 20 sec
```

Port Access

Port 21 is the default SSH or Secure Shell. If this was open to the general public, we could connect to [ip/websiteurl]:21 with a tool like PuTTY. PuTTY is an application to remotely access another machine and execute commands. If we had the username and password we could access the terminal. If you have terminal access you could manipulate the SQL database or upload a script that would do any number of things that like gaining control of the website.

Port 80 is the default web port. We expect people to connect on this and it's not a huge vulnerability.

Password hashing

Firebase incorporates an internal and modified version of scrypt to hash passwords so even if someone can access the hashed passwords, they won't be able to easily apply them.



```
hash_config {
  algorithm: SCRYPT,
  base64_signer_key: jxspr8Ki0RYycVU8zykbdLGjFQ3McFUH0uiiTvC8pVMXAn210wjLNmdZJzxUECKbm0QsEmYUSDzZvpjeJ9WmXA==,
  base64_salt_separator: Bw==,
  rounds: 8,
  mem_cost: 14,
}
```

Future

To improve on this project, the servers both Sad and Good can be improved with more features including but not limited to a review section and a shopping cart. More features of the site will allow for more vulnerabilities and practice with defending from these vulnerabilities.

Shopping Cart

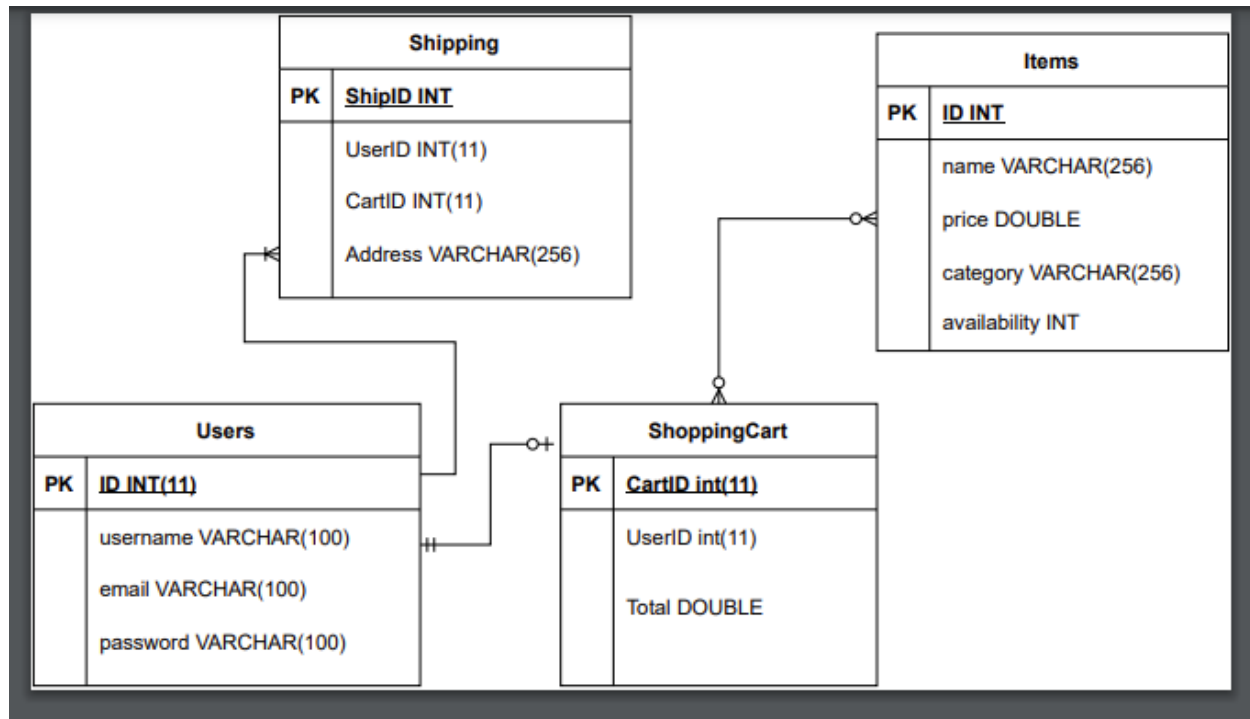
[Deselect all items](#)

		Price
<input checked="" type="checkbox"/> 	Gala Apples Fresh Produce Fruit, 3 LB Bag by GALA APPLES At The Neighborhood Corner Store In Stock Shipped from: Tropical Importers <input type="checkbox"/> This is a gift Learn more Qty: 1 <input type="button" value="Delete"/> <input type="button" value="Save for later"/>	\$18.19
<input checked="" type="checkbox"/> 	Golden State Pears to Compare Deluxe Fruit Gift by Golden State Fruit In Stock ✓prime <input type="checkbox"/> This is a gift Learn more Qty: 1 <input type="button" value="Delete"/> <input type="button" value="Save for later"/> <input type="button" value="Compare with similar items"/>	\$24.93
Subtotal (2 items): \$43.12		

Appendix

Entity Relationship Diagram

This is the structure of our databases on both the Good and Sad Servers.



User Manual

Name:

Hayden Flagg

Team:

Cyber Security

Course:

CS 481 – Capstone Project

Date:

March 6, 2021

Introduction

This user manual is for the purpose of familiarizing readers with the two websites and the hacking software related the cyber security themed software designed by Hayden Flagg, Pierce Peterson, Jesse Villanueva, and Anthony Quintero. The two websites covered in this manual are both mock stores that allow users to login and sign-up as well as look at items that are in the store. The hacking software is made from HTML, JS, and PHP. It is then used through a chrome extension which allows users to hack the website and either view or manipulate the database attached to it. The goal of our software is not to have a great functional website but to try and hack the different servers and note the comparisons and difference in the results of our attempts.

Sad Server

The “sad” server is a simple to use website meant for looking at store items. The users can access the website by going to the login page. From here users can login by entering their information or they can sign up by clicking on the link which can take them to the sign-up page. Like the login page users can then create an account or they can go back to the login page if they have an account. Once a user has created an account or logged in the user will be able to see their information, like their name or how much money they. Besides viewing their personal information users can also view store information like the items in the store. Because the focus of this project is not the website itself but instead the database attached to the website and

how we can hack it, this is the majority of what our website does, and it does not allow for transactions between the store and the customer.

Good Server

The website for the “good” server acts just like the “bad” server’s website. It is a simple website where users can login or signup and then view the information of the user and the items in the store. Users can use this version of the website exactly like they would use the bad server version. The only usable difference between the two versions of the website is that this version allows for users to create a shopping by selecting items to buy. However, like the “sad” version, the “good” server’s website does not allow for transactions.

Hacking Software

Like mentioned earlier, the hacking software that the team made is a chrome extension made from the combination of HTML and JavaScript. The purpose of this software is to allow the user to try and manipulate the two before mentioned servers. After attacking the server, the user should then be able to view or manipulate the server’s data. To do this, users will first have to open the website in Google Chrome. From there users will then need to be in Google Chromes dev mode. While in dev mode the user can load the unpacked extension by selecting the sqlextension folder located in the repository. After doing this the webpage can be reloaded

and will then change to include extra buttons which will have the hacking functions that the user will want. These buttons have different purposes which are described on the button. With these buttons now available, the user can then click a button which should attack the website. After this, if the user's attack was successful then the database for the attacked server should be open and available for the user to see.

Conclusion

Like mentioned before the point of our team's project is to analyze cyber security and ways that websites can be attacked. Therefore, the two websites for our servers are not very functional and only have the purpose of showing information to the user. Our hacking software is also for the very purpose of comparing our two servers, so that we can compare their strengths and weaknesses to better understand how to protect websites.