PT Report

“SuperDuperMarket”

Executive Summary

During our review of SuperDuperMarket's checkout mechanism, our cybersecurity expert conducted a thorough assessment of the website's configuration and identified several vulnerabilities that could be exploited to gain unauthorized access to sensitive documents. Specifically, it was found that the configuration of SuperDuperMarket's servers and environment could be leveraged to gain access to restricted files.

Our cybersecurity expert was able to exploit these vulnerabilities in a specific sequence, allowing him to gain access to a file that should only be accessible to the system administrator.

This vulnerability could enable a semi-experienced attacker to gain unauthorized access to files residing on SuperDuperMarket's server, potentially compromising the functionality of the online store and the security of its clients.

Conclusions

Our professional team perspective in this review, is that the overall security level of SuperDuperMarket's checkout mechanism is **Low**.

SuperDuperMarket's checkout mechanism has been found to be misconfigured in such a way that easily allows one to change the transaction barcode. While this change alone is not a risk, a hacker which knows what he's looking for can take advantage of this and inject a script. Such a script allows to procure sensitive files from a server and in SuperDuperMarket's case our expert was indeed able to gain access to sensitive and otherwise restricted files.
Based on this, the main exploitation vector on SuperDuperMarket's is as follows:

- Unauthorized Access to Sensitive Files via Script Injection

The exploitation of this vulnerability requires a **mid to high** level of skill and knowledge, bringing the team to consider that this vulnerability might be considered as high, but, with the rise of AI, the knowledge barrier to such exploitations is much lower which colours this vulnerability as a definite critical vulnerability.

Finding Details

VULN-001 Unauthorized Access to Sensitive Files via Script Injection

Script injection through PDF mechanism (**Critical**)

Description

Script injection is a type of attack that involves injecting malicious code into a vulnerable website or web application. This can be achieved by exploiting vulnerabilities such as cross-site scripting (XSS), SQL injection, or command injection.

Once the code is “injected”, the malicious code is executed in the victim's browser or server, allowing the attacker to perform various actions such as stealing sensitive data, modifying web content, or redirecting users to malicious sites.

Details

During our review of SuperDuperMarket's checkout mechanism, our cybersecurity expert has identified several vulnerabilities that could be exploited to gain unauthorized access to sensitive files stored on the web server.

First step was to enumerate the website, leading to a discovery that SuperDuperMarket's website had a publicly accessible robots.txt file. Upon further investigation, it was found that the robots.txt file contained a reference to an admin JavaScript API file.

Second step was done through the use of burp suite, a tool that allows interception and modification of traffic packets sent between the client and the server, in this case, during the checkout process. In the checkout process, it was discovered that the transaction barcode parameter was re-writable, leading to a possible script injection attack which was indeed verified to exist.

Third step was to develop a proof-of-concept exploit to demonstrate such an injection impact. Using the script and the data that was uncovered during the OSINT (Open-Source Intelligence) scan, allowed to modify the script to retrieve sensitive data which in this case, meant our cyber expert was able to read and even save the admin JS API file.

This vulnerability allows gaining unauthorized access to sensitive files stored on SuperDuperMarket's web server. If exploited by an attacker, this vulnerability could result in significant financial, reputational, and legal damages to SuperDuperMarket and its customers.

Evidence

In such cases the first step is OSINT, looking to understand what the website is compromised of, users on it, what it runs and so on. During this step our cyber expert has discovered that the common path of ‘robots.txt’ path exists and open for user navigation

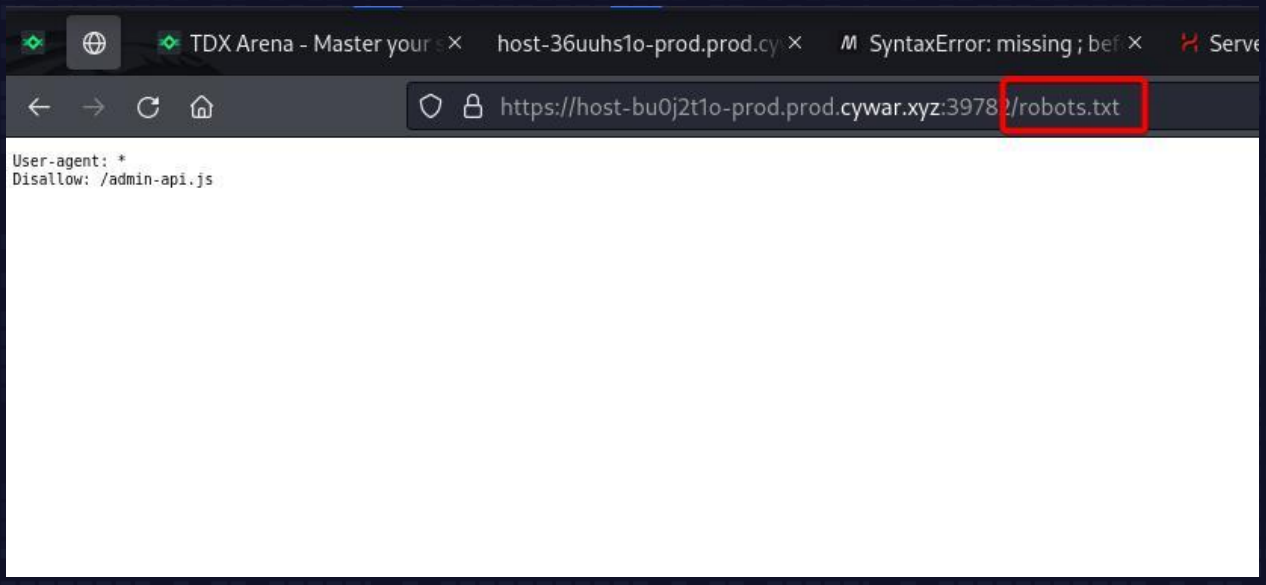

FIGURE 1: ROBOTS.TXT IS ADDED AS A PATH AND IS INDEED ACCESSIBLE

Once the ‘robots.txt’ path has been accessed an interesting file path is revealed, an ‘admin-api.js’ file which based on the name suggests we found a sensitive java script file used by admin to do some actions, maybe as an API

FIGURE 2: ADMIN-API.JS FILE AS SHOWN BY THE ROBOTS.TXT PATH

Applying the new ‘admin-api.js’ path to SuperDuperMarket's store path

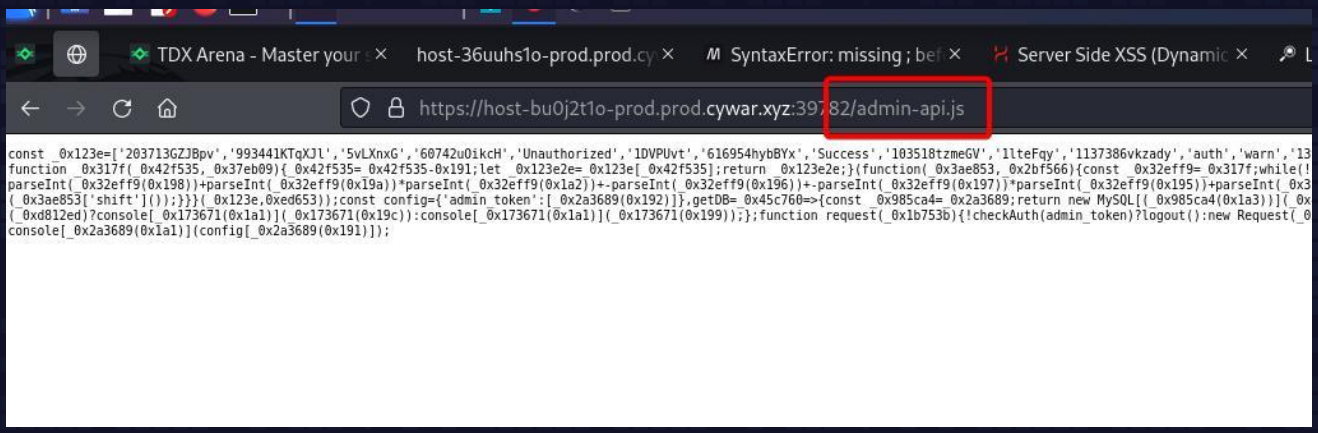

FIGURE 3: ADMIN-API.JS FILE HAS BEEN SET AS THE PATH TO NAVIGATE TO AND IS ACCESSIBLE

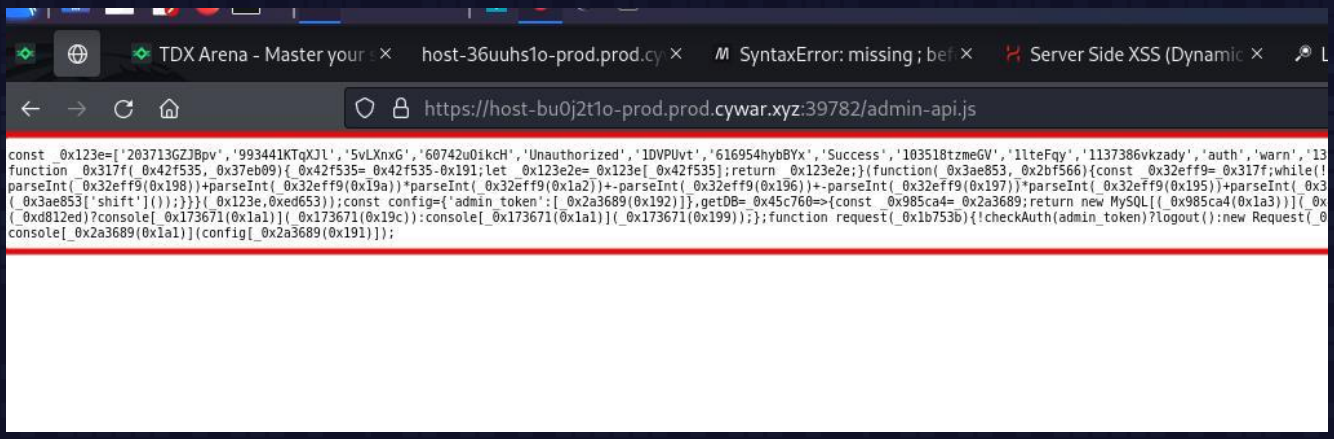

FIGURE 4: ADMIN-API.JS FILE CONTENTS IS EXPOSED TO A PUBLIC USER

Now burp comes into play to act as a bridge that permits or halts any traffic sent between the client and SuperDuperMarket's website. Once turned to intercept mode, any packet moving between client and server will be captured and stopped

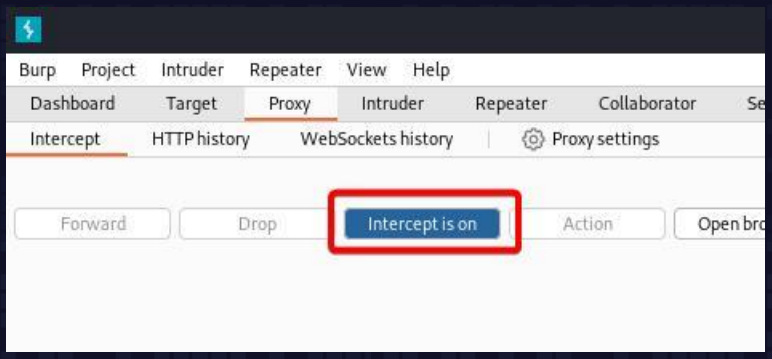

FIGURE 5: BURP SUITE IS RUNNING AND INTERCEPTS THE TRAFFIC

Proceeding to make a purchase, once the transaction goes through several confirmations it is intercepted and stopped by burp, allowing full packet analyzation and modification

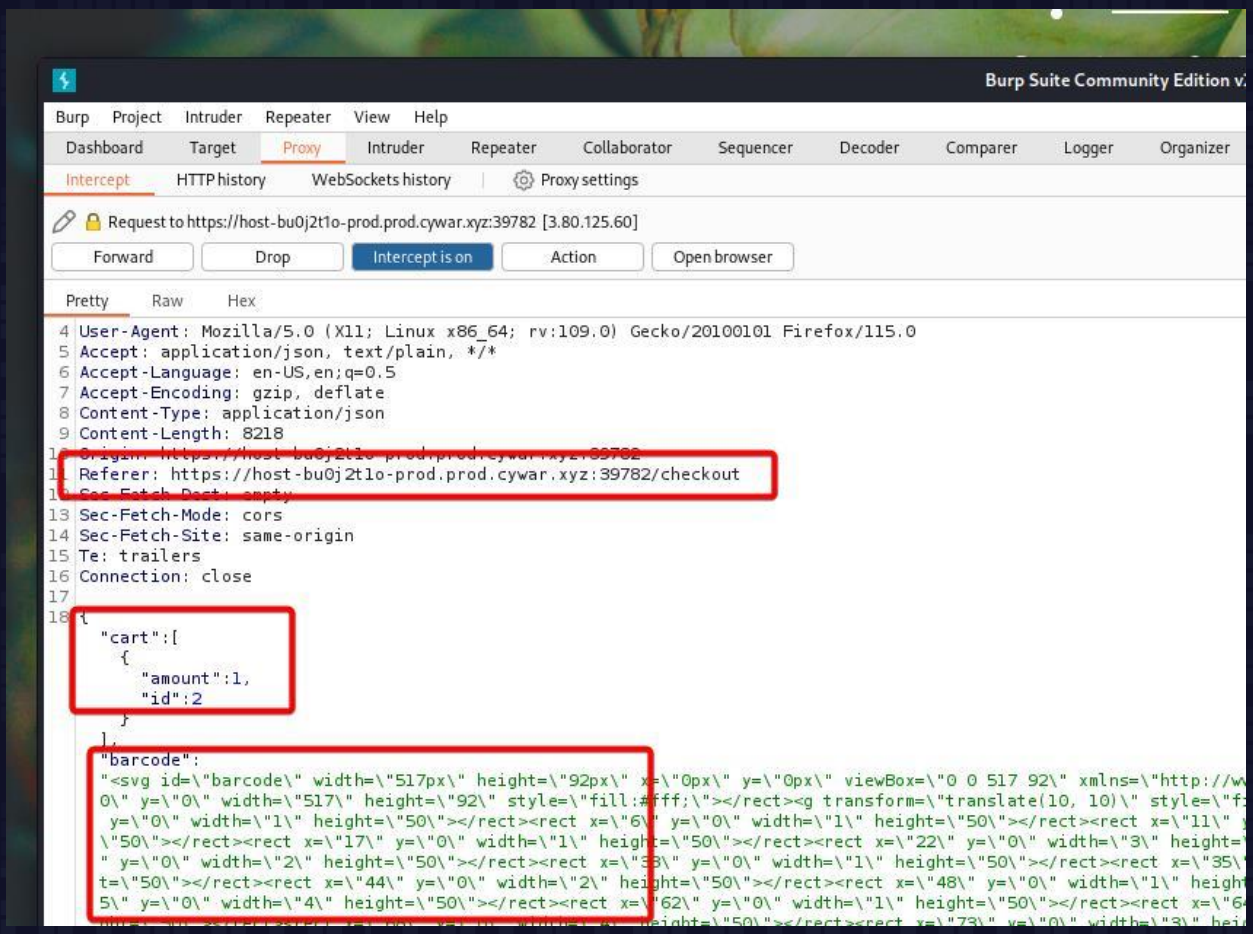

FIGURE 6: TRANSACTION PACKET IS ANALYZED, SHOWING WEBSITE ADDRESS, CONTENTS, BARCODE AND MORE

Analyzing the packet structure a commonly known target is recognized, the employed barcode method, the SVG barcode technique. After some trial and error our expert honed in on the barcode digits themselves. As POC, the barcode is replaced with HTML tags

FIGURE 7: INTERCEPTED BARCODE DIGIT SECTION IS IDENTIFIED

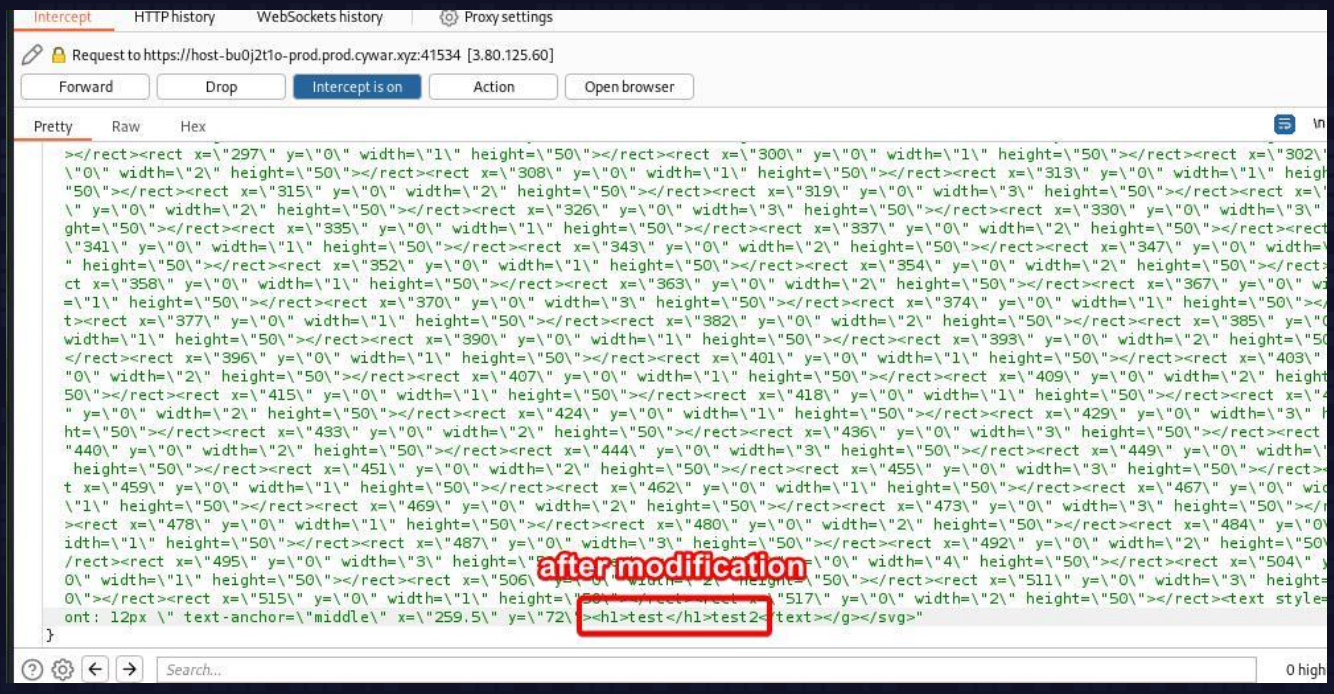

FIGURE 8: BARCODE DIGITS CHANGED TO CHECK FOR HOW HTML PAYLOAD WORKS ON SUPERDUPERMARKET'S RECEIPT

The modified packet is now released back to the webserver with the edited SVG, a simple preliminary payload

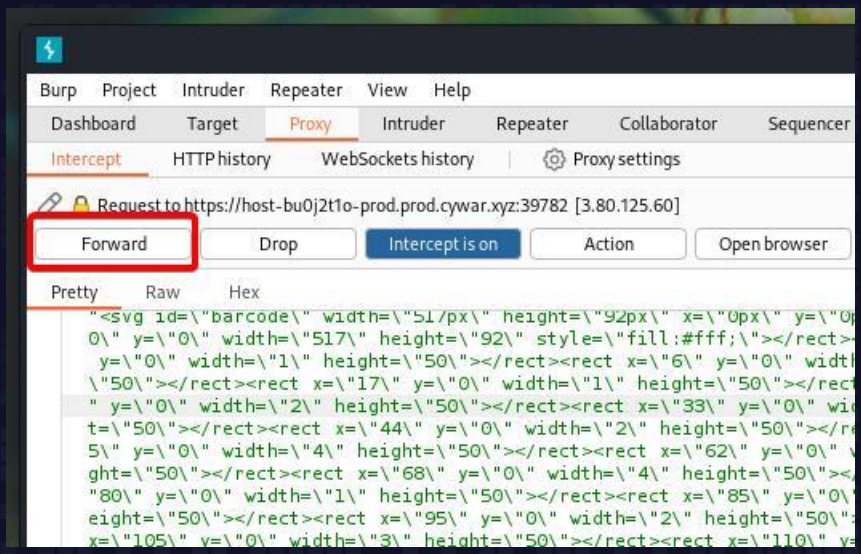

FIGURE 9: BURP SUITE – THE EDITED PACKET / TRANSACTION IS SENT BACK TO THE WEBSITE

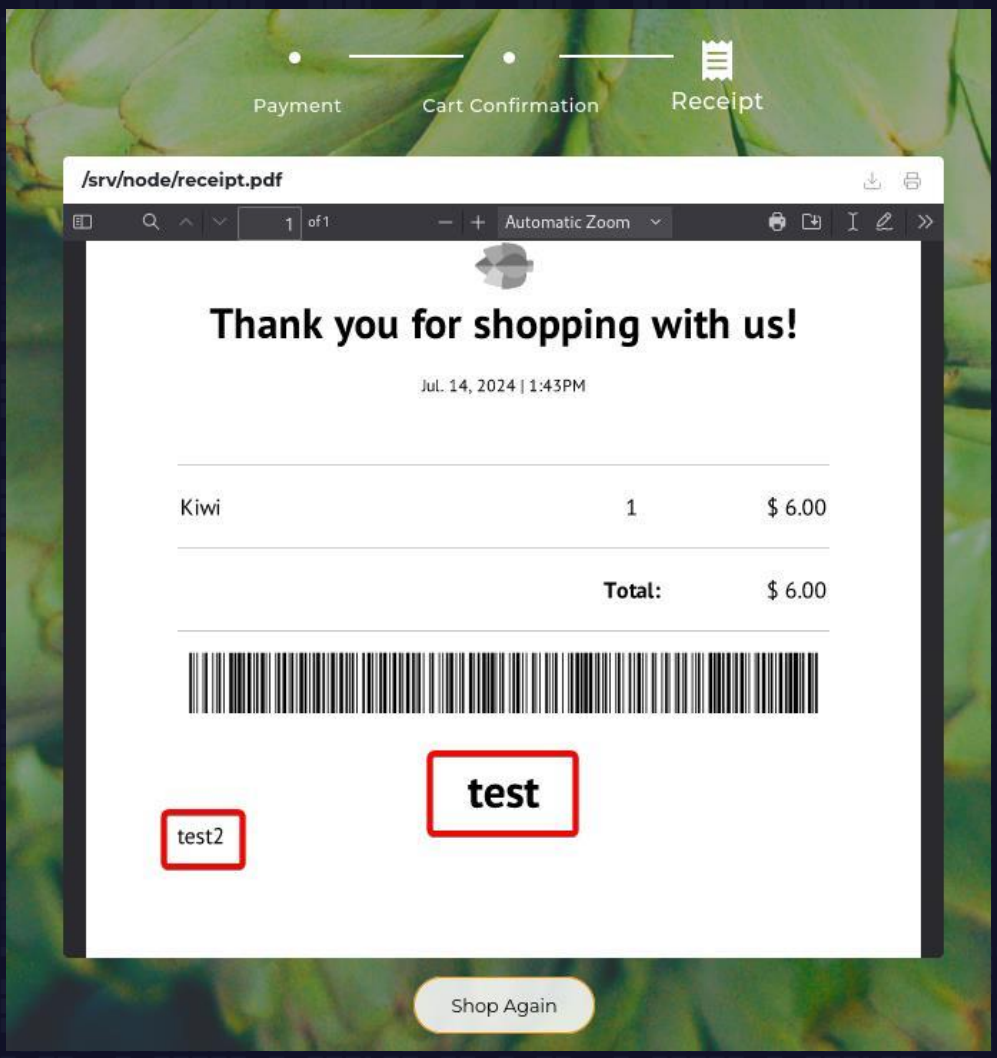

FIGURE 10: THE PDF GENERATION TOOL REACTS TO HTML TAGS AND EMBEDS IT IN THE SCRIPT

As the HTML injection worked, a new specific payload found and matched to SuperDuperMarket's vulnerability

The buying process is repeated, using the new payload where the goal now is to access and retrieve sensitive info from SuperDuperMarket's server

```
<script>
    x=new XMLHttpRequest;
    x.onload=function(){
        document.write(this.responseText)
    };
    x.open("GET","file:///etc/passwd");
    x.send();
</script>
```

FIGURE 11: THE PAYLOAD THAT WAS MATCHED TO FIT SUPERDUPERMARKET'S VULNERABILITY

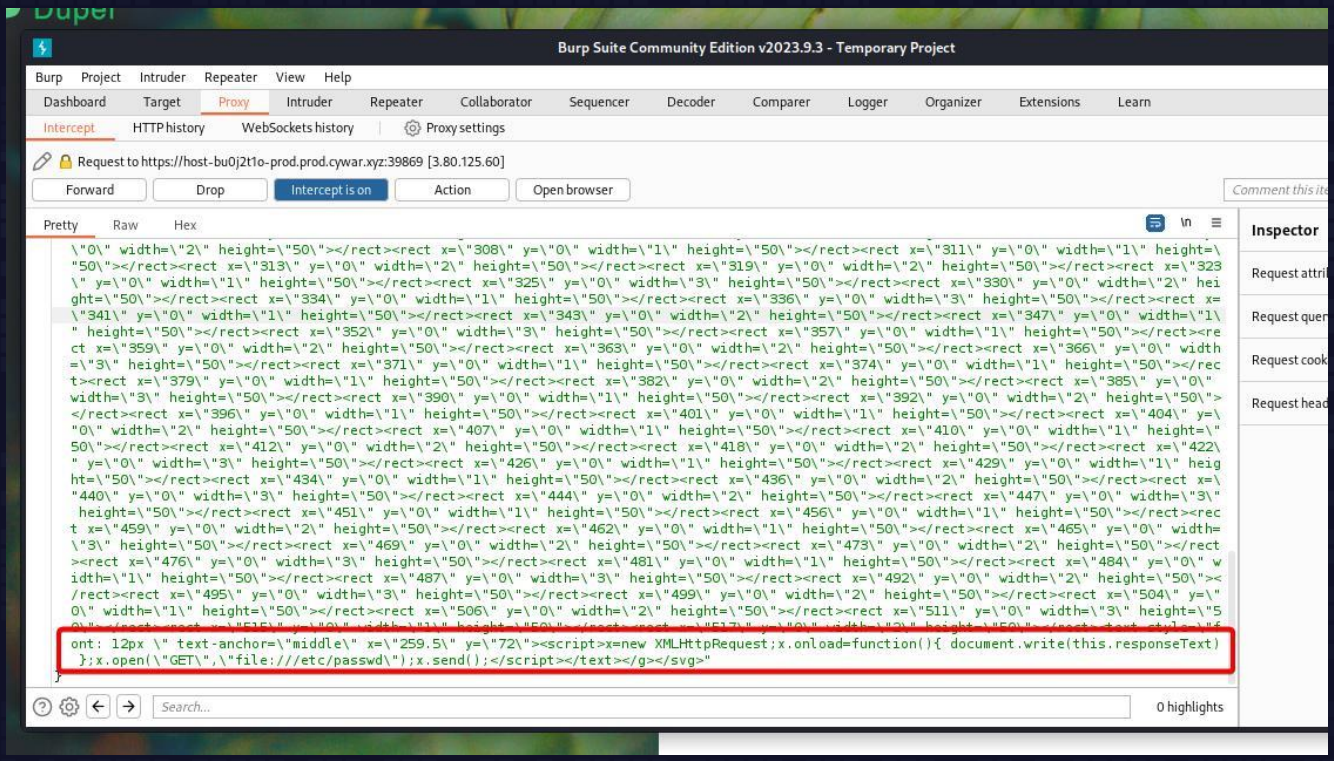

FIGURE 12: THE TRANSACTION IS LOADED WITH THE SCRIPT PAYLOAD NOW AIMING TO RETRIEVE A SENSITIVE FILE

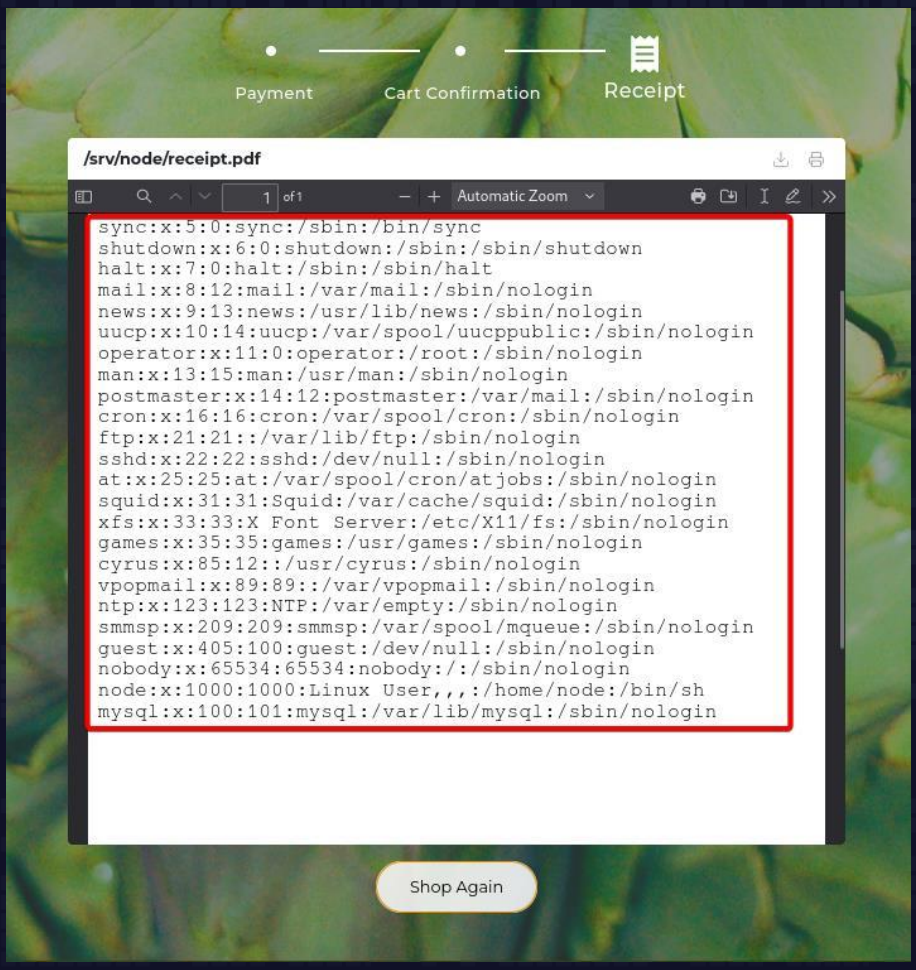

FIGURE 13: ALL THE USERS IN THE SYSTEM ARE DISPLAYED ON THE CLIENT RECEIPT

Now our expert combines the OSINT knowledge discovered in previous steps, with the ability to access sensitive files that are part of the website functionality

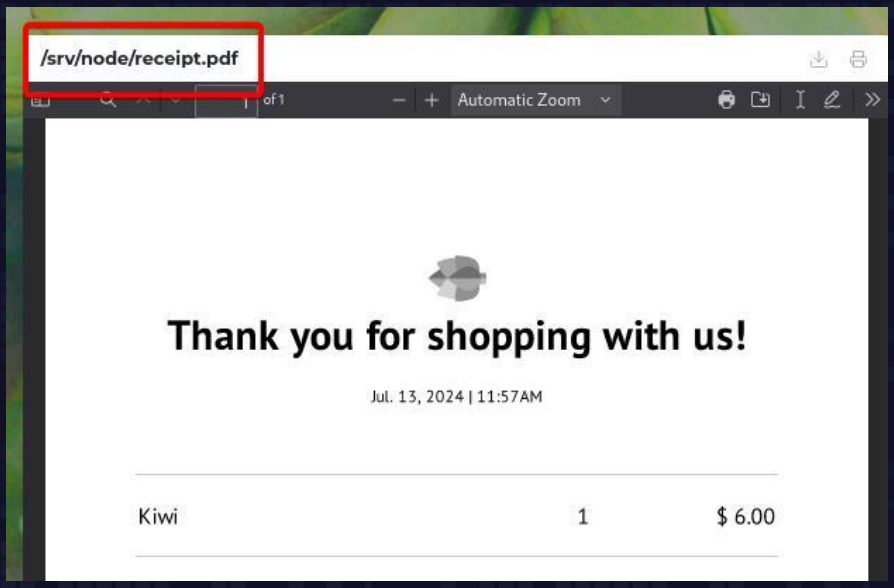

FIGURE 14: PATH ON SERVER IS REVEALED ON THE RECEIPT PAGE AND AIDS IN FINDING THE TARGET PATH

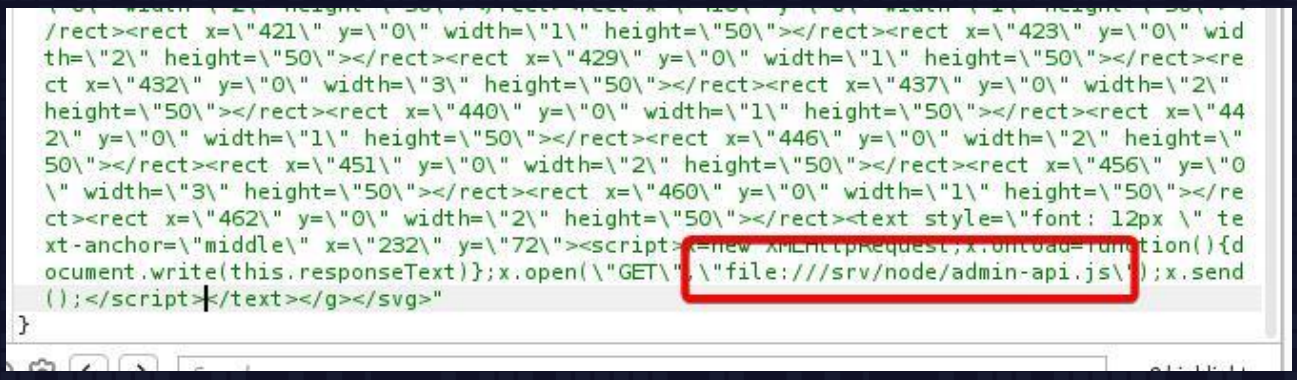

FIGURE 15: SCRIPT PAYLOAD EDITED TO PULL THE ADMIN-API.JS FILE WITH THE HELP OF THE EXPOSED SERVER PATH

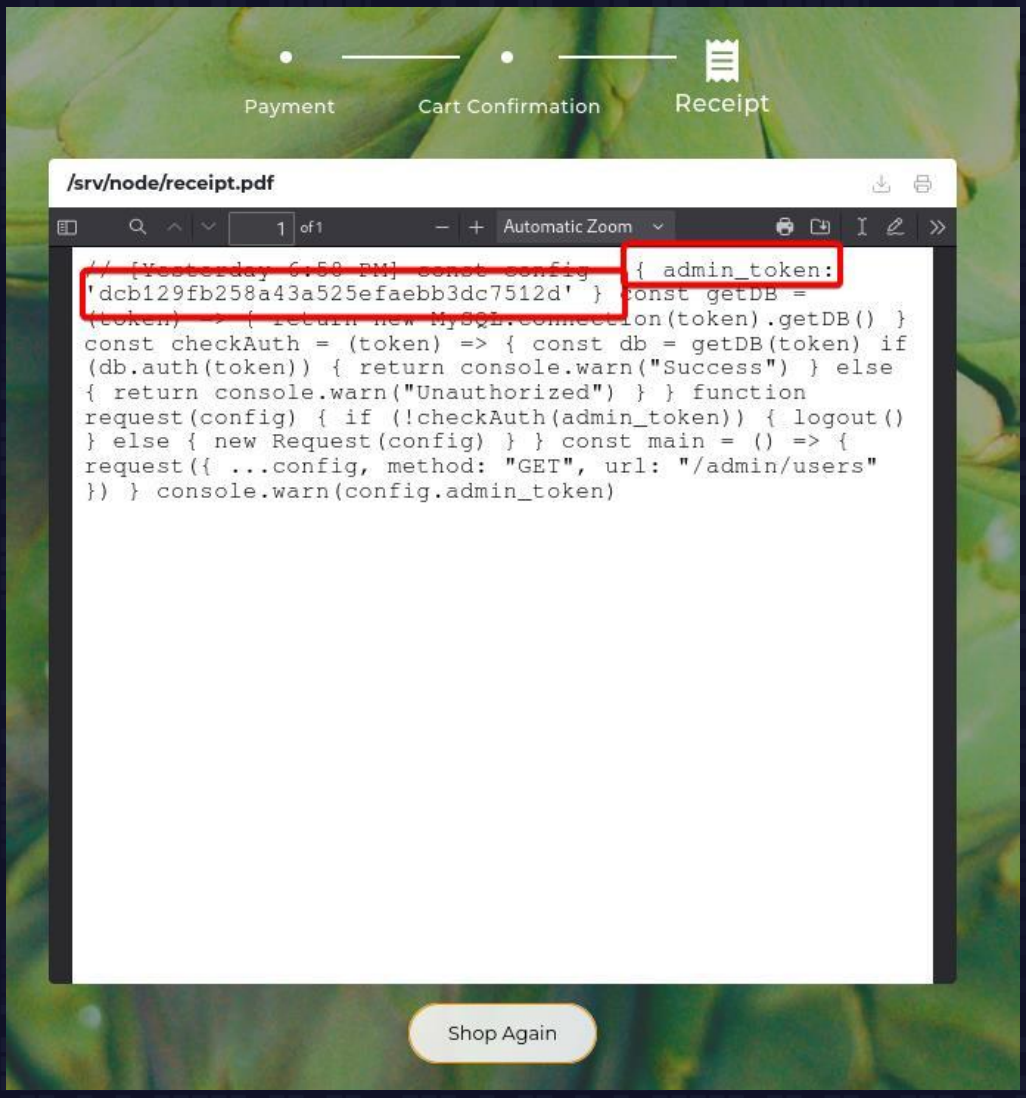

FIGURE 16: THE ADMIN TOKEN IS SEIZED AND CAN NOW BE USED FOR FURTHER EXPLOITATION

Remediation Options

- It is recommended to implement authentication and authorization when accessing the ‘robots.txt’ public path. Only authorized users should have access to this path.
  Note: it is recommended to monitor any additional public paths that may be added in the future and make sure they are also allowed for authorized users only.
  Note2: This may prevent search engines from accessing this file in which case the SuperDuperMarket website can try either switching to use the ‘noindex’ meta tag or ‘X-Robots-Tag’ HTTP header or adopt the below sub-bullet as a suggested alternate method altogether.
  - In case blocking access to ‘robots.txt’ is not a viable option, it is recommended to remove sensitive paths from publicly accessible files such as ‘robots.txt’ to prevent attackers from discovering sensitive data through OSINT techniques. As an immediate step it is recommended to remove the admin JavaScript file path from the publicly accessible robots.txt file.
- It is recommended to adopt the principle of least privilege (PoLP) for the PDF process – By creating a specific process to run the PDF generation, and restricting it to the minimum access it needs to perform its duty, it will no longer be exploitable in the manner demonstrated here.
  Note: It is recommended to adopt this approach for each and every user or service which has access to SuperDuperMarket's server / system.
- It is recommended to sanitize all inputs in the PDF Generation process - Ensure that any input included in the PDF generation is sanitized to prevent script injection.
- It is recommended to make sure that any data that is presented to the client doesn’t include sensitive info such as the server path as seen on the receipt page. As an immediate step it is recommended to remove the server path from the user’s receipt page view.
- It is recommended to Implement a web application firewall (WAF) to detect and block malicious inputs.