Lifestyle Store – A Web Application

Detailed Developer Report

Security Status – Extremely Vulnerable

- Hacker can steal all records in Internshala databases (SQLi)
- Hacker can take control of complete server including View, Add, Edit, Delete files and folders (Shell Upload)
- Hacker can change source code of application to host malware, phishing pages or even explicit content (Shell Upload)
- Hacker can inject client side code into applications and trick users by changing how page looks to steal information or spoil the name of Internshala (XSS)
- Hacker can extract mobile number of all customers using Userid (IDOR)

Vulnerability Statistics

Critical	Severe	Moderate
7	9	11

Low
5

Vulnerabilities:

No	Severity	Vulnerability	Count
1	Critical	SQL Injection	3
2	Critical	Access to Admin panel	1
3	Critical	Arbitrary File Upload	1
1	Critical	Account takeover via OTP Bypass	1
5	Severe	cross site scripting	1
6	Critical	CSRF	1
7	Severe	Reflected Cross Site Scripting	1
8	Severe	Components with known vulnerability	3
9	Moderate	Server Misconfiguration	1
10	Severe	Brute Force	1

Vulnerabilities:

No	Severity	Vulnerability	Count
11	Severe	Forced Browsing	2
12	Moderate	Client Side Filter Bypass	1
13	Severe	Default/Common Password	1
14	Moderate	IDOR-Unauthorized access to user details	4
15	Moderate	Directory Listing	5
16	Low	Personal Information Leakage	2
17	Low	Default Message	1
18	Low	Open Redirection	2

1. SQL Injection

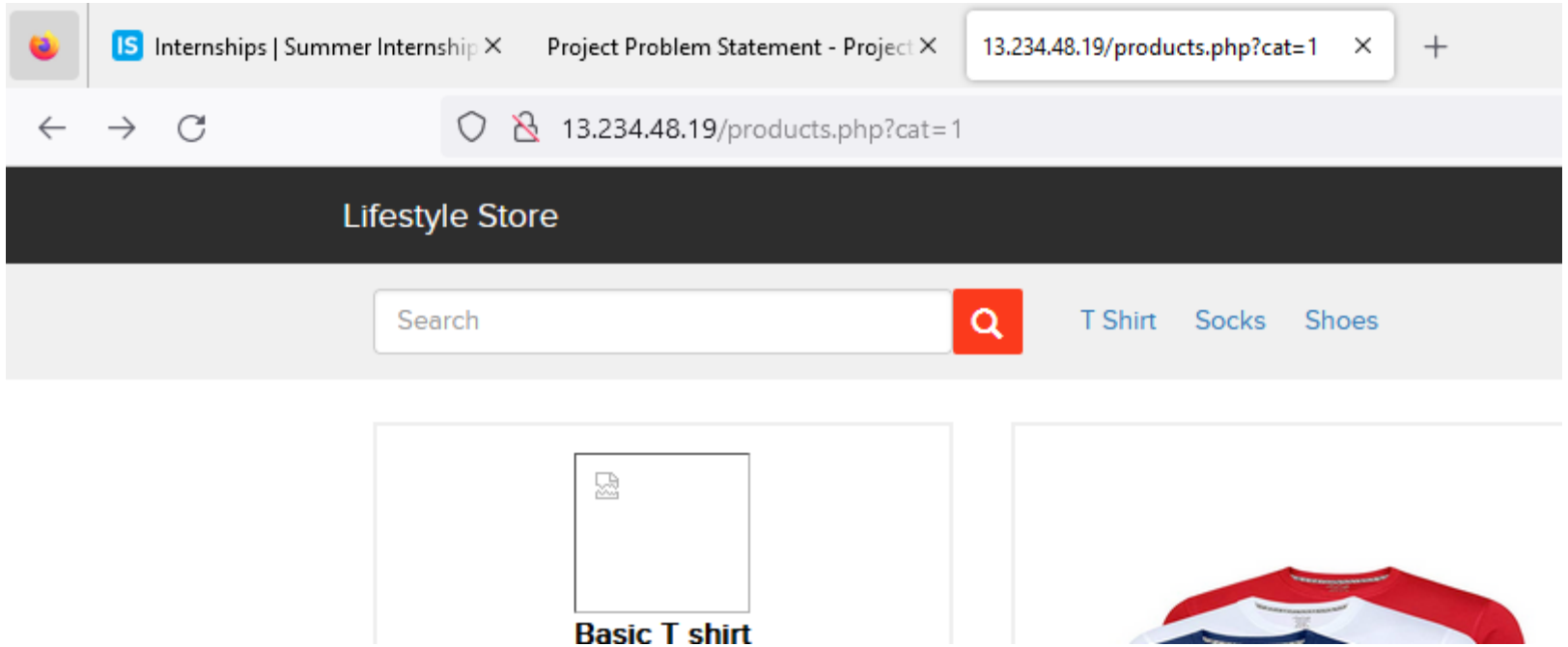
SQL Injection (Critical)	Below mentioned URL in the **Lifestyle Store – An Ecommerce Website** is vulnerable to SQL injection attack **Affected URL :** • http://13.234.48.19/products.php?cat=1 **Affected Parameters :** • cat (GET parameter) **Payload:** • Cat = 1’

1. SQL Injection

SQL Injection (Critical)	Here are other similar SQLi in the application **Affected URL :** • http://13.234.48.19/products.php?cat=2 • http://13.234.48.19/products.php?cat=3

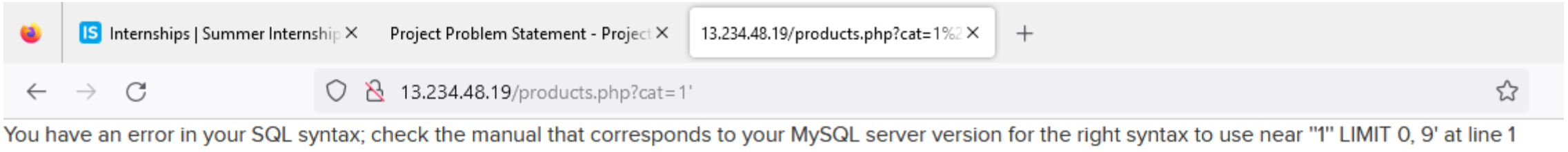
Observation

- Navigate to the Main Page of the website where you will see categories option click on **T Shirt** or **Socks** or **Shoes** to get into this URL, you will see products as per the category you have chosen, but notice the **GET parameter** in the URL. It shows cat=1 means that T shirts has assigned cat id as 1



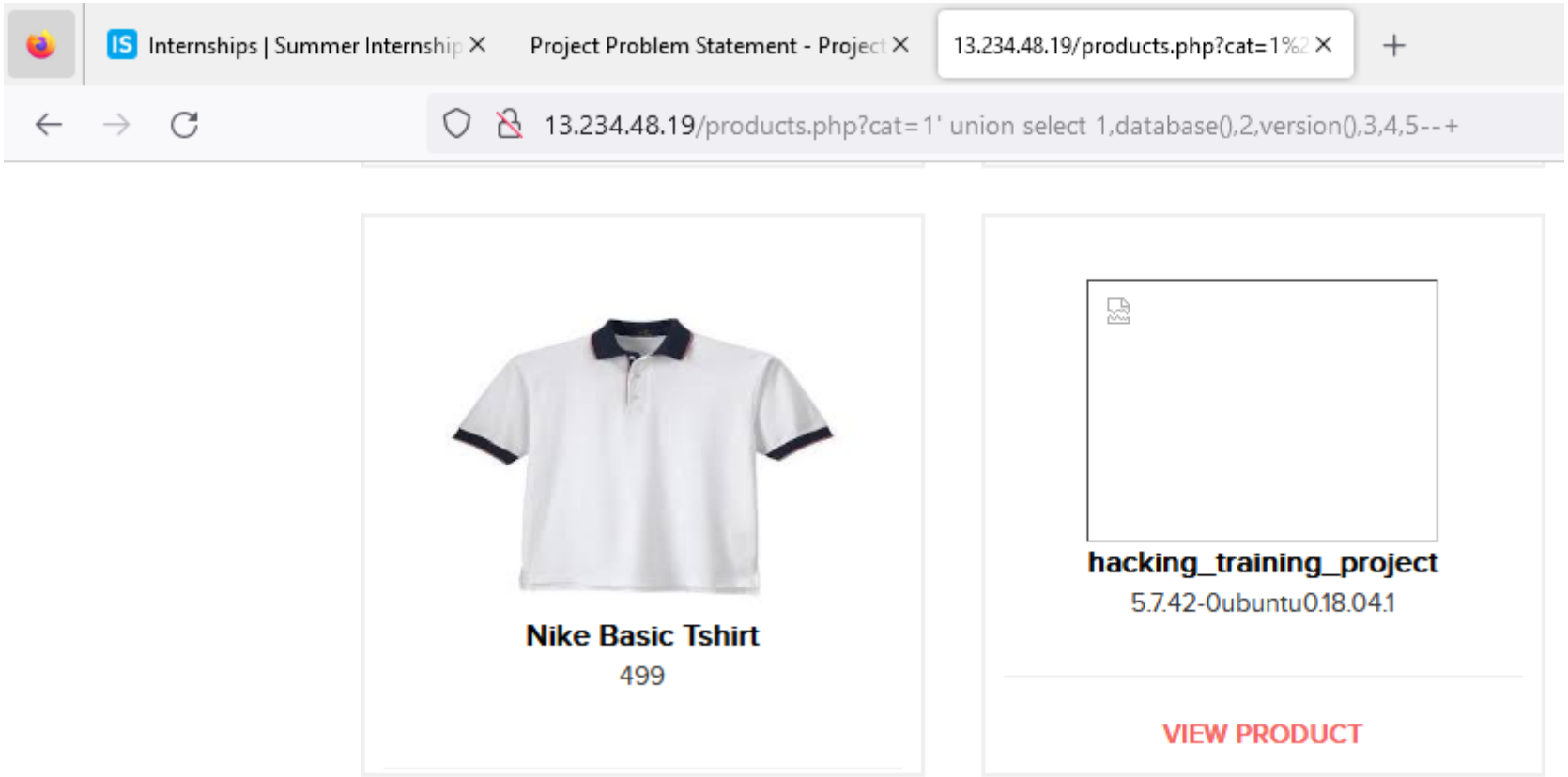
Observation

- Now, we apply **single quote** in category parameter(i.e. GET parameter): **http://13.234.48.19/products.php?cat=1’** and we get complete **MySQL error.**



Proof of Concept (PoC)

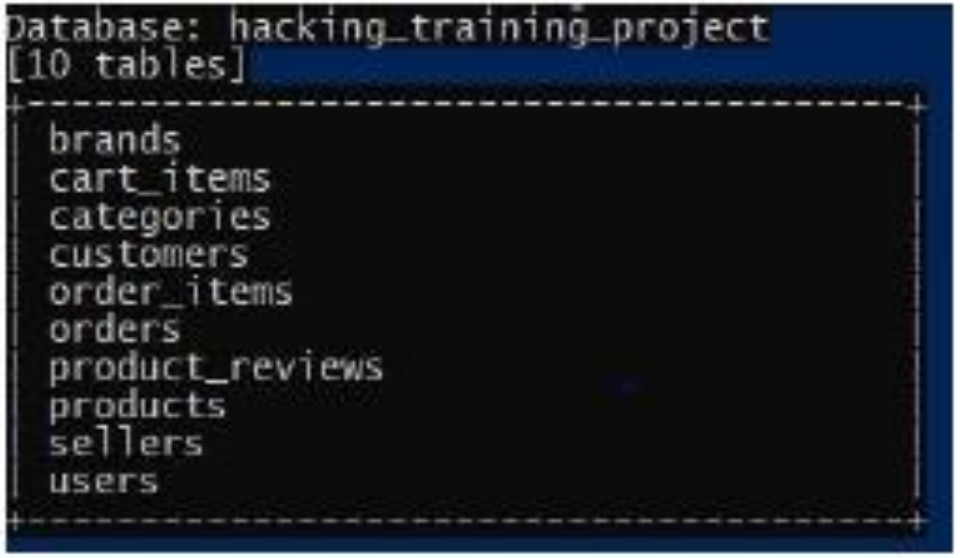
- Attacker can execute SQL commands as shown below. Here we have used the payload below to extract the database name and MySQL version information: **http://13.234.48.19/products.php?cat=1' union select 1,database(),2,version(),3,4,5--+**



PoC – Attacker can dump arbitrary data

- No of databases: 2
  - Information_schema
  - Hacking_training_project

- No of tables in hacking_training_project: 10
  - brands
  - Cart_items
  - Categories
  - Customers
  - Order_items
  - Orders
  - Products
  - Sellers
  - Users
  - Product_reviews



Business Impact – Extremely High

- Using this vulnerability, attacker can execute arbitrary SQL commands on Lifestyle store server and gain complete access to internal databases along with all customer data inside it.
- Below is the screenshot of users table which shows user credentials being leaked, although the password is encrypted yet vulnerable and can be misused by hackers.
- Attacker can use this information to login to admin panels and gain complete admin level access to the website which could lead to complete compromise of the server and all other servers connected to it.

Database: hacking_training_project
Table: users
[15 entries]

user_name	password	phone_number	unique_key
admin	$2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	8521479630	15468927955c66694cba1174.29688447
Donal234	$2y$10$PM.7nB5P5FMaldXiM/S3s./p5xR6GTKvjry7ysJtxOkBqOJURAHsO	9489625136	778522555c6669996f5a24.34991684
Pluto98	$2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	8912345670	19486318945c666a037b1432.99985767
chandan	$2y$10$4cZBEIrgthXdvT1hwUlivuFELeO3rR.GIcdpO3NjrlSOveiOKLVDa	7854126395	12404594545c666a3b49e0f8.08173871
Popeye786	$2y$10$Fkv1RfwYTioWOw2CaZtAQuXVnhGAUjt/If/yTqkNPC5zTrsVm7EeC	9745612300	18430379145c666a53af8431.79566371
Radhika	$2y$10$RYxNhOyV/G4g7OtFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC.	9512300052	15611262655c666b312f73e0.70827297
Nandan	$2y$10$G.cRNLMEiG79ZFXE1Hg.R.o95334UOxmZu4.9MqzR5614ucwnk59K	7845129630	1587354115c666b65bb44a5.36505317
MurthyAdapa	$2y$10$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1DPR.11UubDG	8365738264	16357203785c68f640c699a2.83646347
john	$2y$10$GhDB8h1X6XjPMY12GZ1vDO7Y3en97u1/.oXTZLmYqB6F18FBgecvG	6598325015	9946437385c6a435f76bef0.14675944
bob	$2y$10$kiUikn3HPFbuyTtK751LNurxzqC0LX3eMGy0/Ux16JOoG37dCGKLq	8576308560	4305822125c6a43ec507df0.68309267
jack	$2y$10$z/nyNlkRJ76m9ItMZ4N51OeRxy6Gkqi9N/UBcJu5ZeO7eM7N4pTHu	9848478231	15257114565c6a444692b707.17903432
bulla	$2y$10$HT5oiRMetqaZ7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG	7645835473	18292501185c6a4493a5ddb0.87138000
hunter	$2y$10$pB3U9iFxwBgSbl2AkBpiEeIBdhiYfwy9y.xV23q12gGbMCyn7N3g2	9788777777	13824560345c80704e821145.26019698
asd	$2y$10$At5pFZnRWpjCD/yNnJWDL.L3Cc4Cv0W8Q/WEHmWzBFqVIkBQFpCF2	9876543210	8057400125c862a7f5916c9.06111587
acdc	$2y$10$J50B78.gpucuLTwpHwbcPedYcain.Yi.tsTLyQtK17FzdSpmIRRbi	9999999999	13104802695c86f43f0c3705.77019309

Recommendation

Take the following precautions to avoid exploitation of SQL injections:

- Do not run Database Service as admin/root user
- Assign each Database user only the required permissions and not all permissions
- Use whitelists, not blacklists
- Don’t trust any user input
- Adopt the latest technologies
- Ensure Errors are Not User Facing
- Disable/remove default accounts, passwords and databases

References

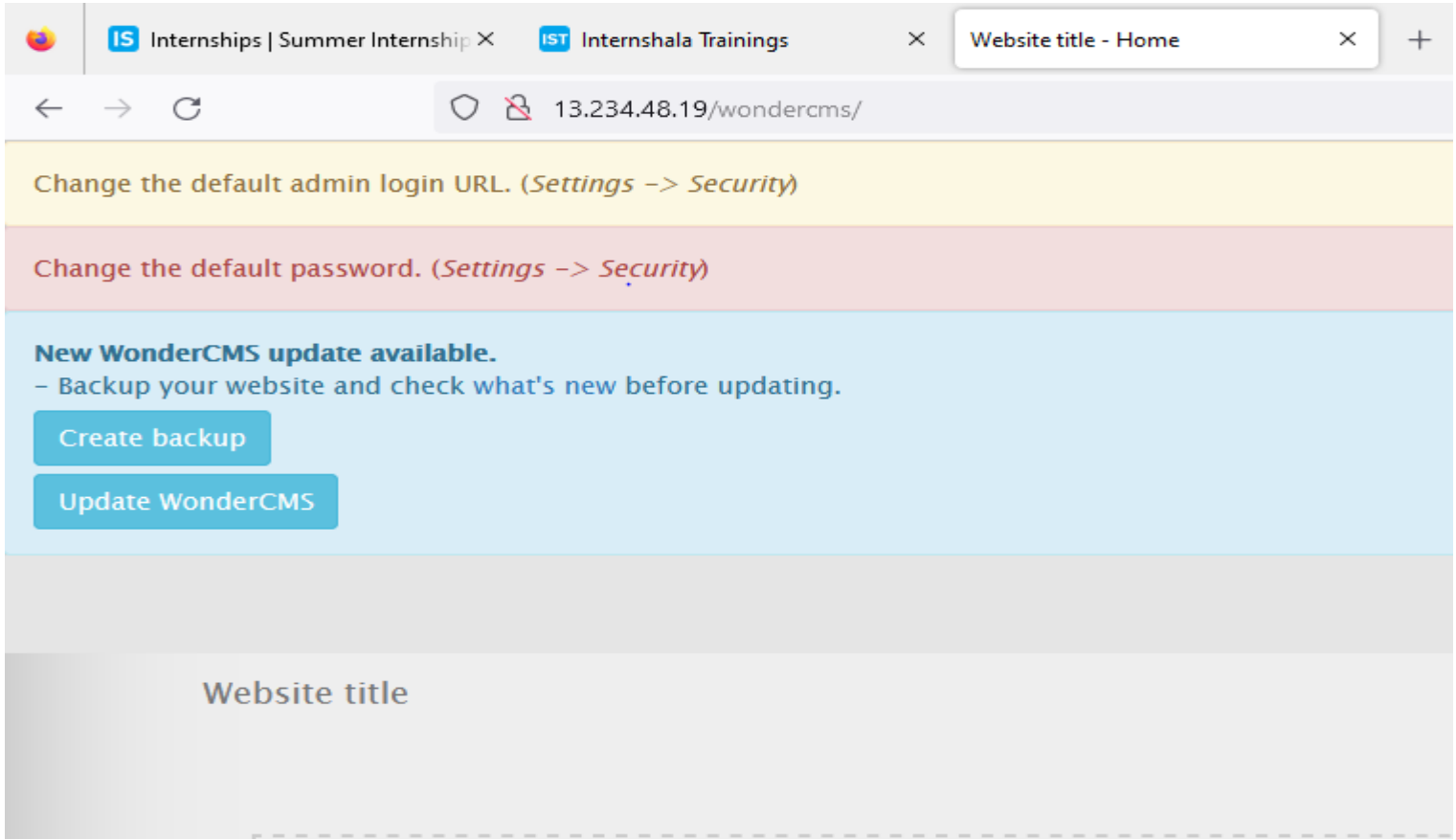
- https://www.owasp.org/index.php/SQL_Injection
- https://en.wikipedia.org/wiki/SQL_injection

2. Access to Admin Panel

Access to Admin Panel (Critical)	Below mentioned URL is vulnerable to **Arbitrary File Upload and making other admin level changes.** **Affected URL :** • http://13.234.48.19/wondercms/loginURL

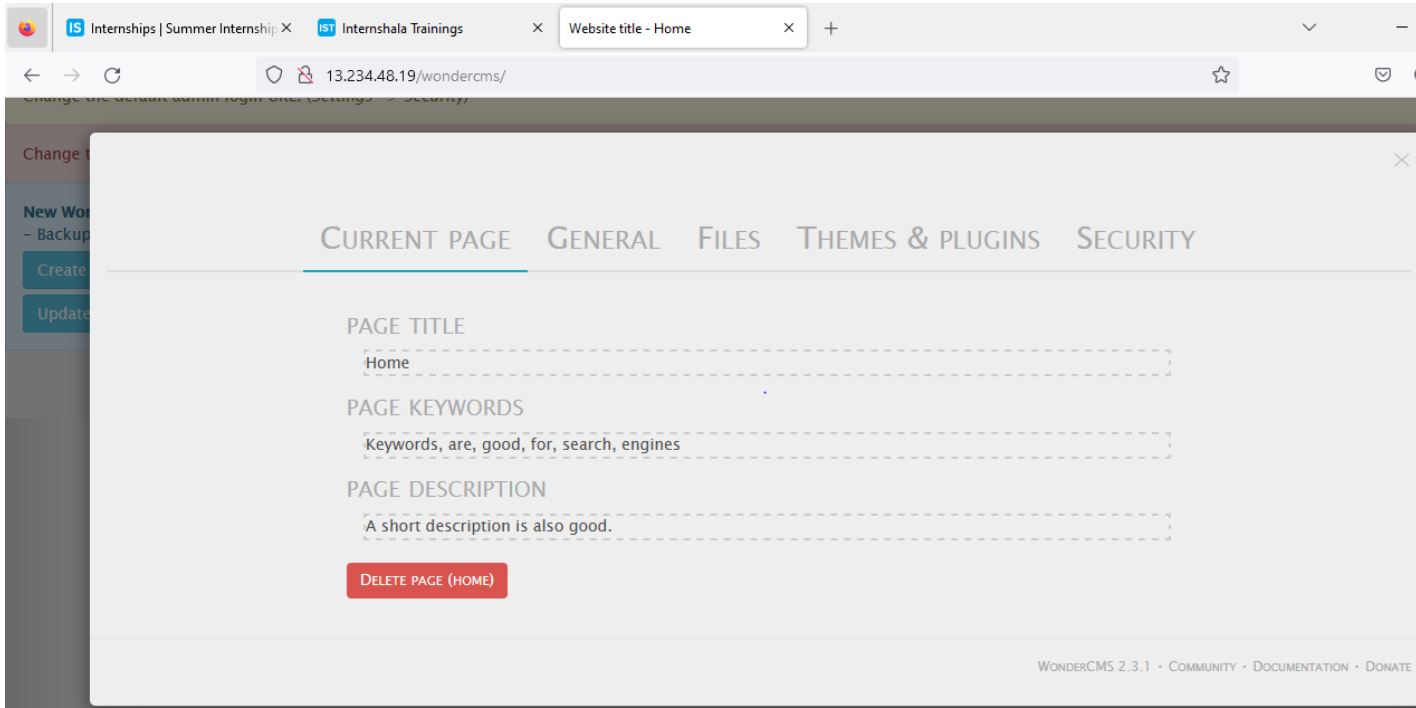
Observation

- When we navigate to http://13.234.48.19/wondercms/loginURL
- we get the password on the page and login as : admin
- When typing admin in the input field we get logged in to the admin panel.



Proof Of Concept - POC

- Attacker can change the admin password .
- Attacker can upload any malicious file.
- Attacker can change page’s title and description.
- Attacker can also make changes in the default theme.
- Attacker can also add and delete pages.



Business Impact – Extremely High

Attacker can do anything with the page, he will have full access of the page and can govern the page according to it's will.

- It is the massive business risk.
- Loss can be very high [Financial as well as Reputational]
- Data and customers trust is compromised.

Recommendation

Take the following precautions:

- Use a strong password 8 character or more in length with alpha-numerics and symbols
- Disable default accounts and users
- Change all passwords to strong unique passwords
- The admin url must also be such that its not accessible to normal users.
- Password changing option must be done with 2 to 3 step verification.

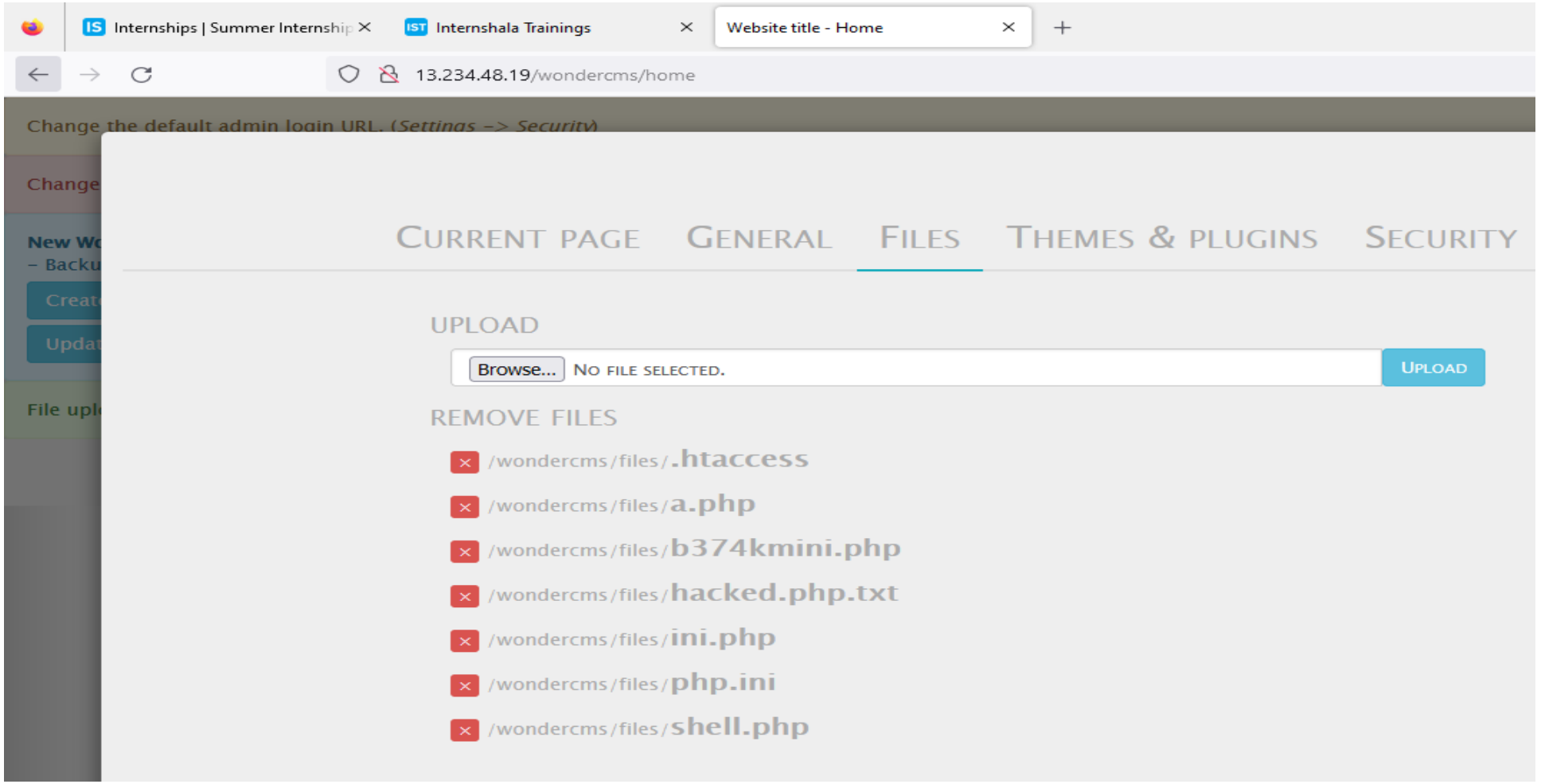
References:

https://www.owasp.org/index.php/Testing_for_weak_password_change_or_reset_functionalities_(OTG-AUTHN-009)
https://www.owasp.org/index.php/Default_Passwords
https://www.us-cert.gov/ncas/alerts/TA13-175A

3. Arbitrary File Upload

Arbitrary File Upload (Critical)	The attacker can upload insecure shells and files and gain access over the entire database and login as the admin and the version is known to have vulnerabilities. **Affected URL :** • http://13.234.48.19/wondercms/home **Affected Parameters :** • File Upload (POST parameters)

Observation



Proof Of Concept - POC

- The password is set as Admin which is a default as well as a week password.
- The password allows the attacker to gain access to admin panel.
- The file upload section is vulnerable to arbitrary file inclusion which allows the attacker to upload backdoors to the website.

Business Impact – Extremely High

A malicious user can access the Dashboard which discloses many critical

information of organization including:

- Important files
- Password
- Changing Website theme
- Tampering with website details and descriptions.
- Any backdoor file or shell can be uploaded to get access to the uploaded file on remote server.

Recommendation

Change the Admin password to something strong and not guessable.

The application code should be configured in such a way, that it should block uploading of malicious files extensions such as exe/ php and other extensions with a thorough server as well as client validation. CVE ID allocated:CVE-2017-14521.

References

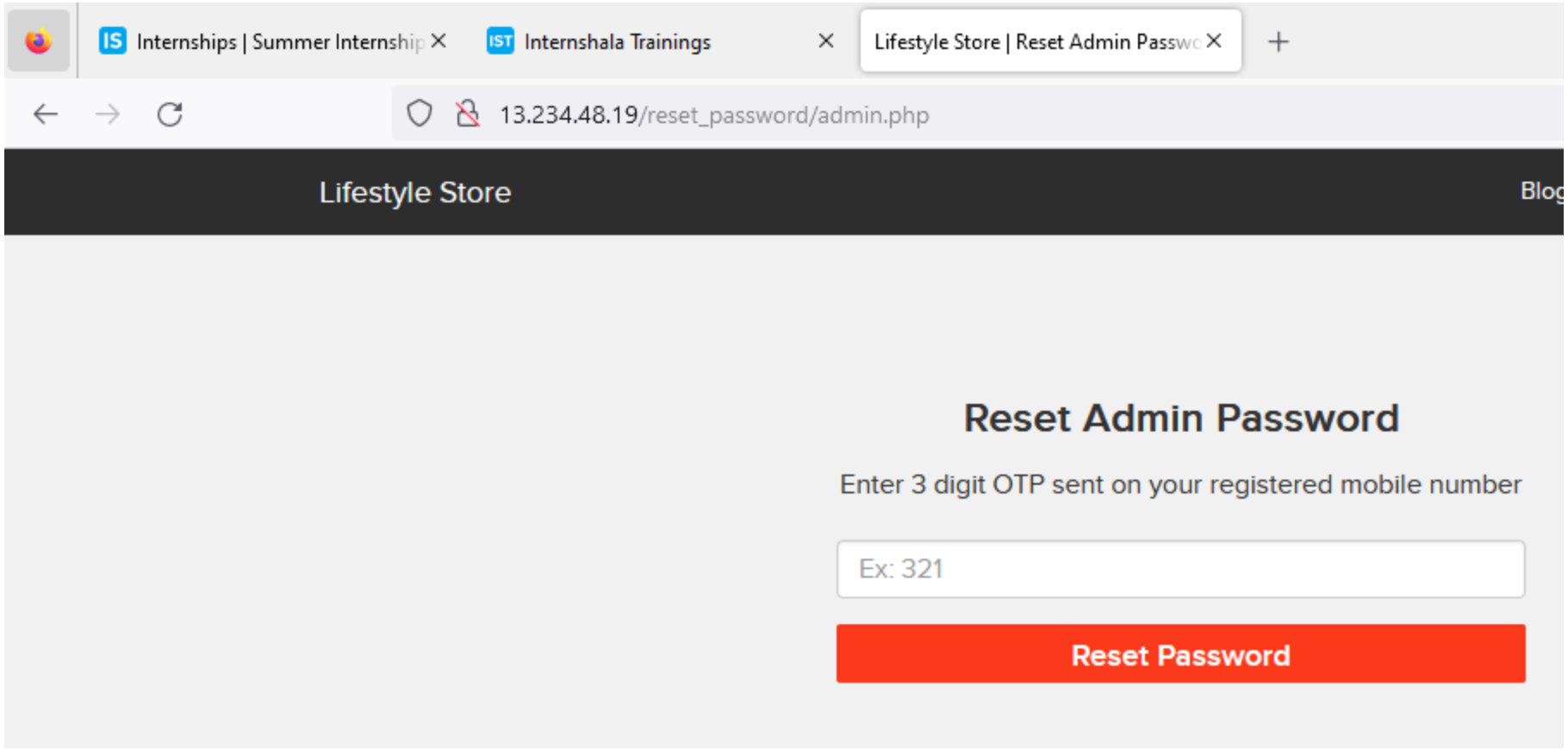
- https://www.owasp.org/index.php/Unrestricted_File_Upload
- https://www.opswat.com/blog/file-upload-protection-best-practices

4. Account Takeover Using OTP Bypass

Account Takeover Using OTP Bypass (Critical)	The below mentioned login page allows login via OTP which can be bruteforced **Affected URL :** • http://13.234.48.19/reset_password/admin.php **Affected Parameters :** • OTP (POST parameters)

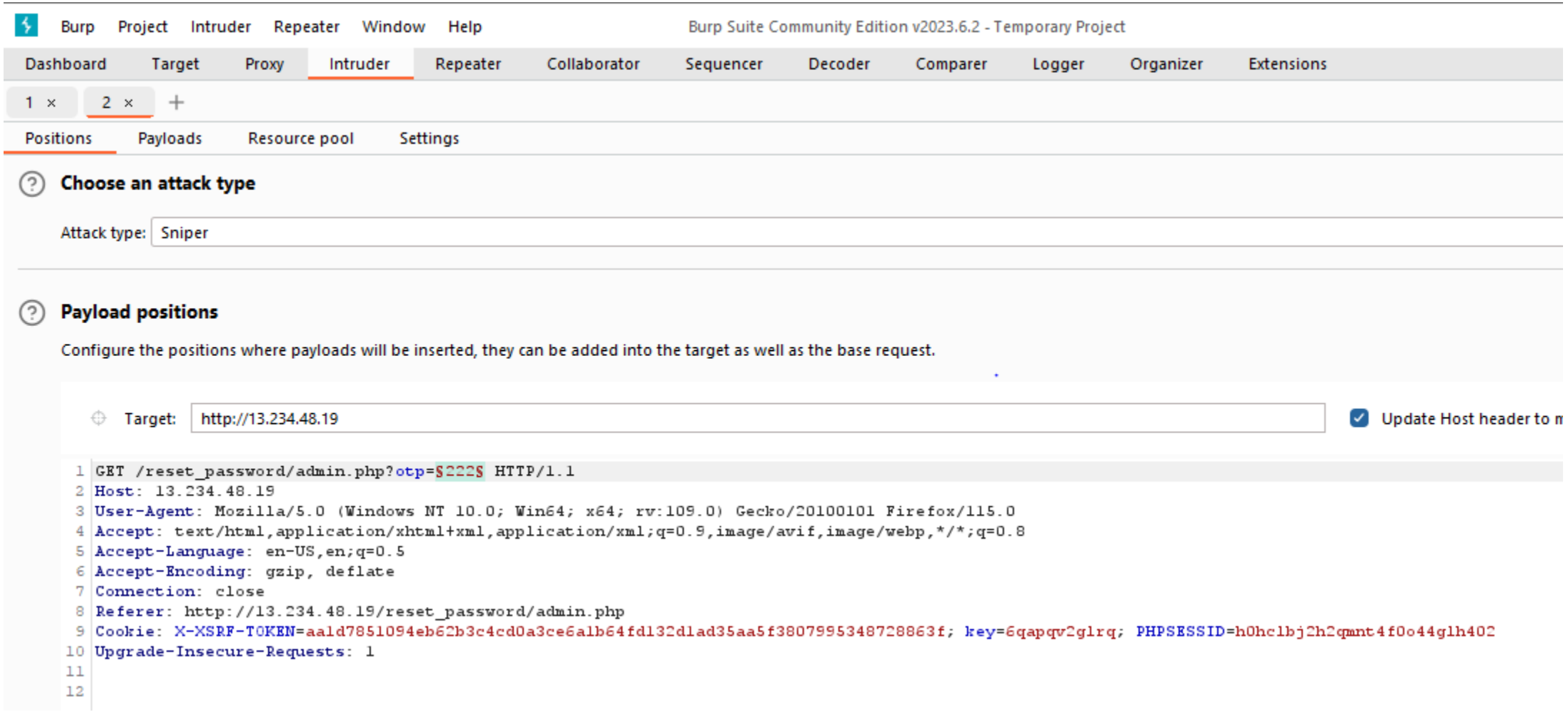
Observation

- Navigate to http://13.234.48.19/reset_password/admin.php You will see user login page via OTP.



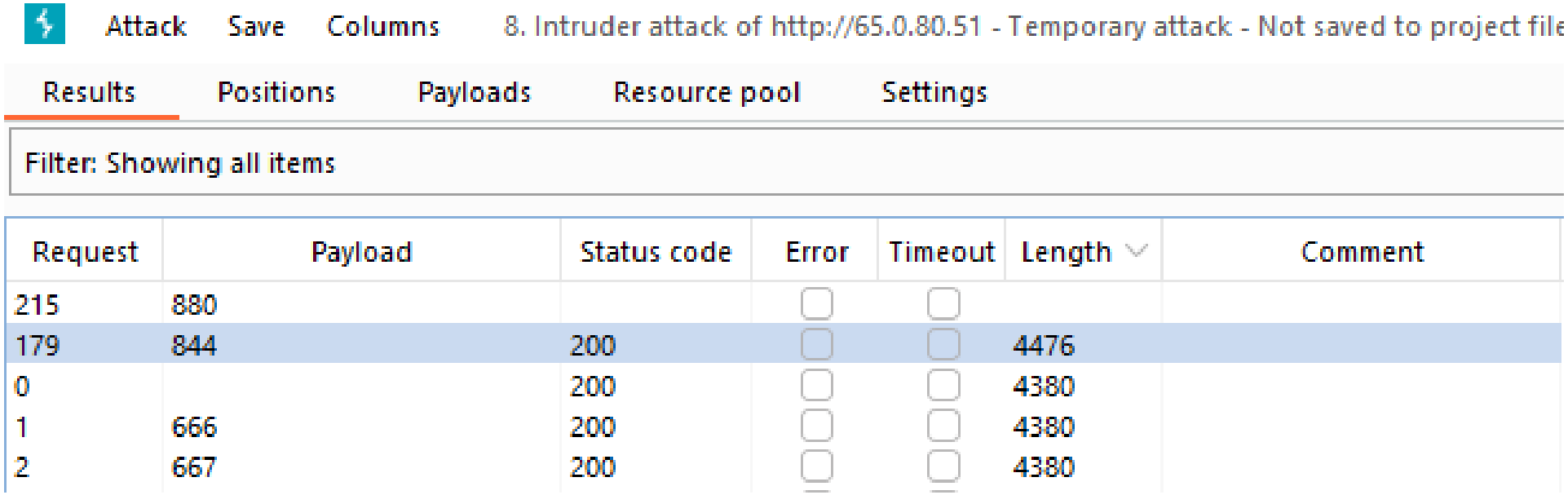
Observation

- Following request will be generated containing OTP parameter.

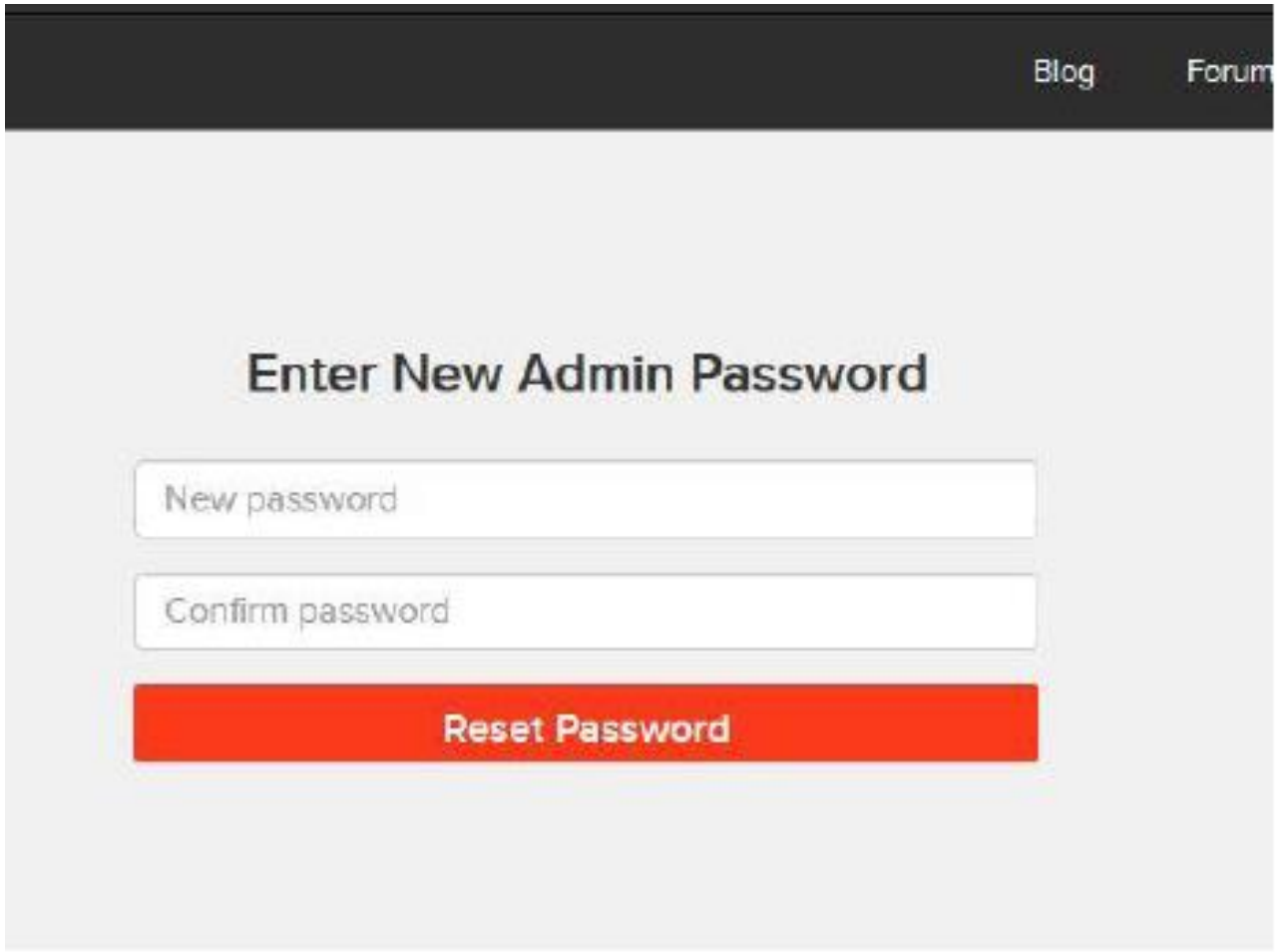


Observation

Here we have easily got the OTP by brute forcing through Burpsuite



Proof Of Concept - POC



Business Impact – Extremely High

- A malicious hacker can gain complete access to any account just by brute forcing the otp.
- This leads to complete compromise of personal user data of every customer.
- Attacker once logs in can then carry out actions on behalf of the victim which could lead to serious financial loss to him/her.

Recommendation

Take the following precautions:

- Use proper rate-limiting checks on the no of OTP checking and Generation requests
- Implement anti-bot measures such as ReCAPTCHA after multiple incorrect attempts
- OTP should expire after certain amount of time like 2 minutes
- OTP should be at least 6 digit and alphanumeric for more security

References:

https://www.owasp.org/index.php/Testing_Multiple_Factors_Authentication_(OWASP-AT-009)
https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks

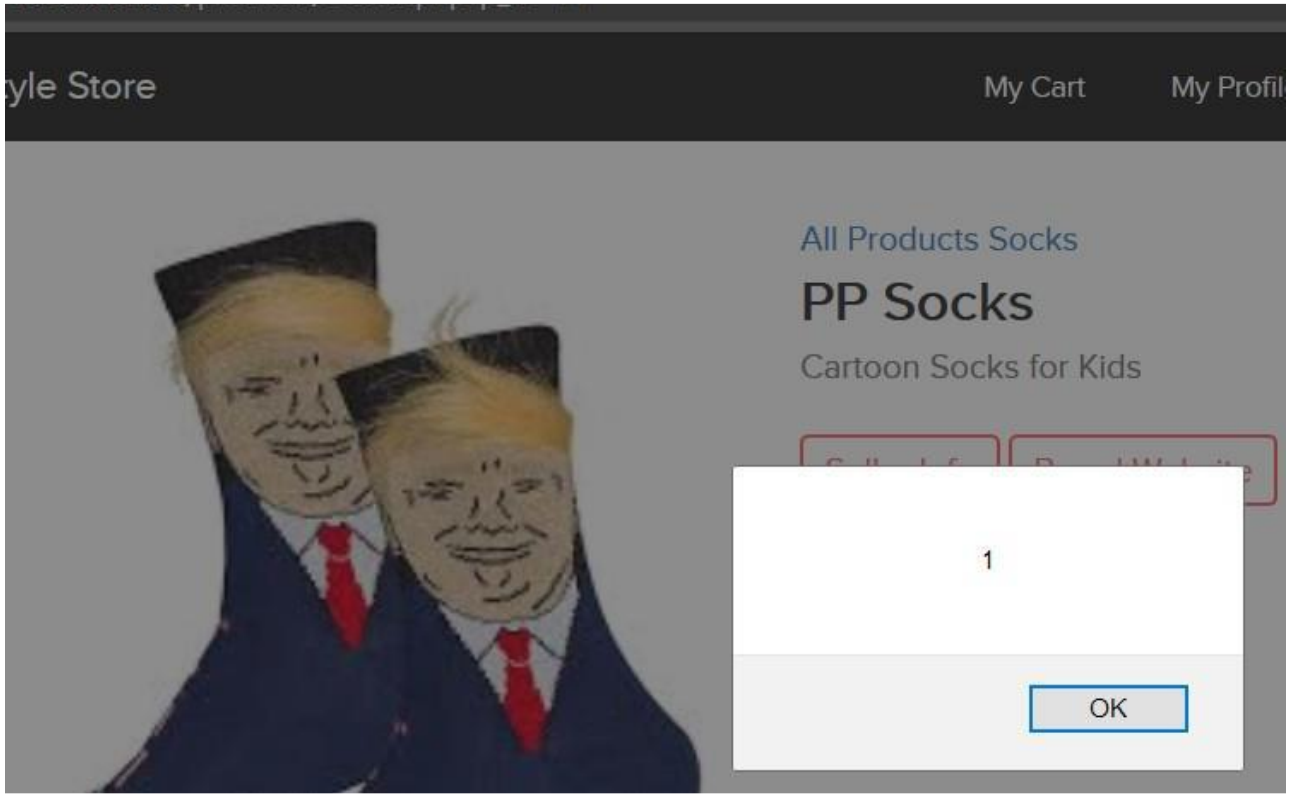
5. cross site scripting

Cross Site Scripting (Critical)	Below mentioned parameters are vulnerable to stored XSS, **Affected URL :** •http://43.205.120.102/products/details.php?p_id=(***all id's***) **Affected Parameters :** •customer review text field **Payload:** •<script>alert(1)</script>

Observation

- Log in to your account. Then go to **My Cart** and then click on **SHOP NOW**button and select any product,
- Or Navigate to http://43.205.120.102/products/details.php?p_id=15 (here I selected product number 11).

POC


yle Store
My Cart
My Profil
All Products Socks
PP Socks
Cartoon Socks for Kids
1
OK

Business Impact - High

- As attacker can inject arbitrary HTML CSS and JS via the review text field, attacker can put any content on the page like phishing pages, install malware on victim's device and even host explicit content that could compromise the reputation of the organization.

```
<div class="review_item">
    <div class="profile_picture">
        <img src="/static/images/uploads/customers/default.png">
    </div>
    <div class="review_details">
        <div class="profile_name">
            <p>anonymous</p>
        </div>
        <div class="profile_review_content">
            <p><script>alert(1)</script></p>
        </div>
    </div>
</div>
```

- All the attacker needs to do is to type in the malicious script in the review field and then anyone opening the link can be attacked by the hacker and victim would see hacker controlled content on the website. As the user trusts the website, he/she will trust the content too.

Recommendation

Take the following precautions:

- Sanitize all user input and block characters you do not want.
- Convert special HTML characters like ‘ “ < > into HTML entities " %22 < > before printing them on the website.

References

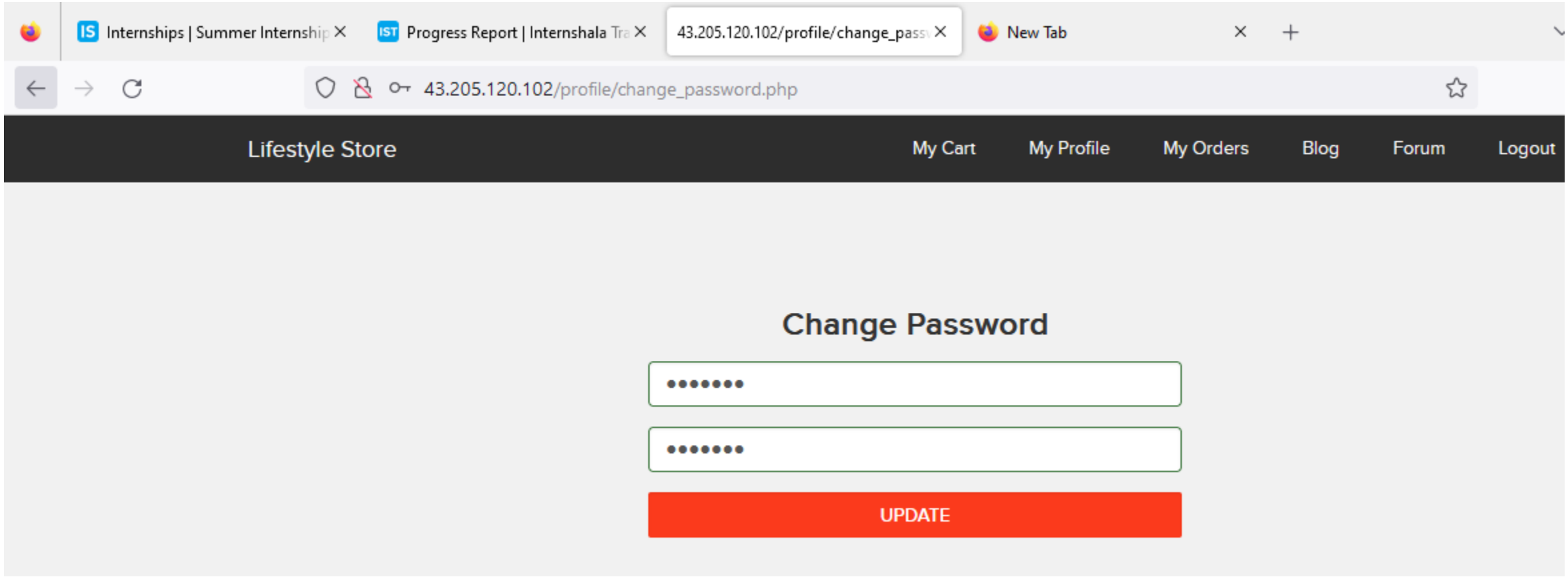
- https://owasp.org/www-community/attacks/xss/
- https://en.wikipedia.org/wiki/Cross-site_scripting
- https://www.w3schools.com/html/html_entities.asp

6. CSRF

CSRF (Critical)	The below mentioned login page allows you to change password without verification and view detailsof other customers (CSRF). Affected URL : •http://43.205.120.102/profile/change_password.php Affected Parameters : •Update button (POST parameter) We can change the password.

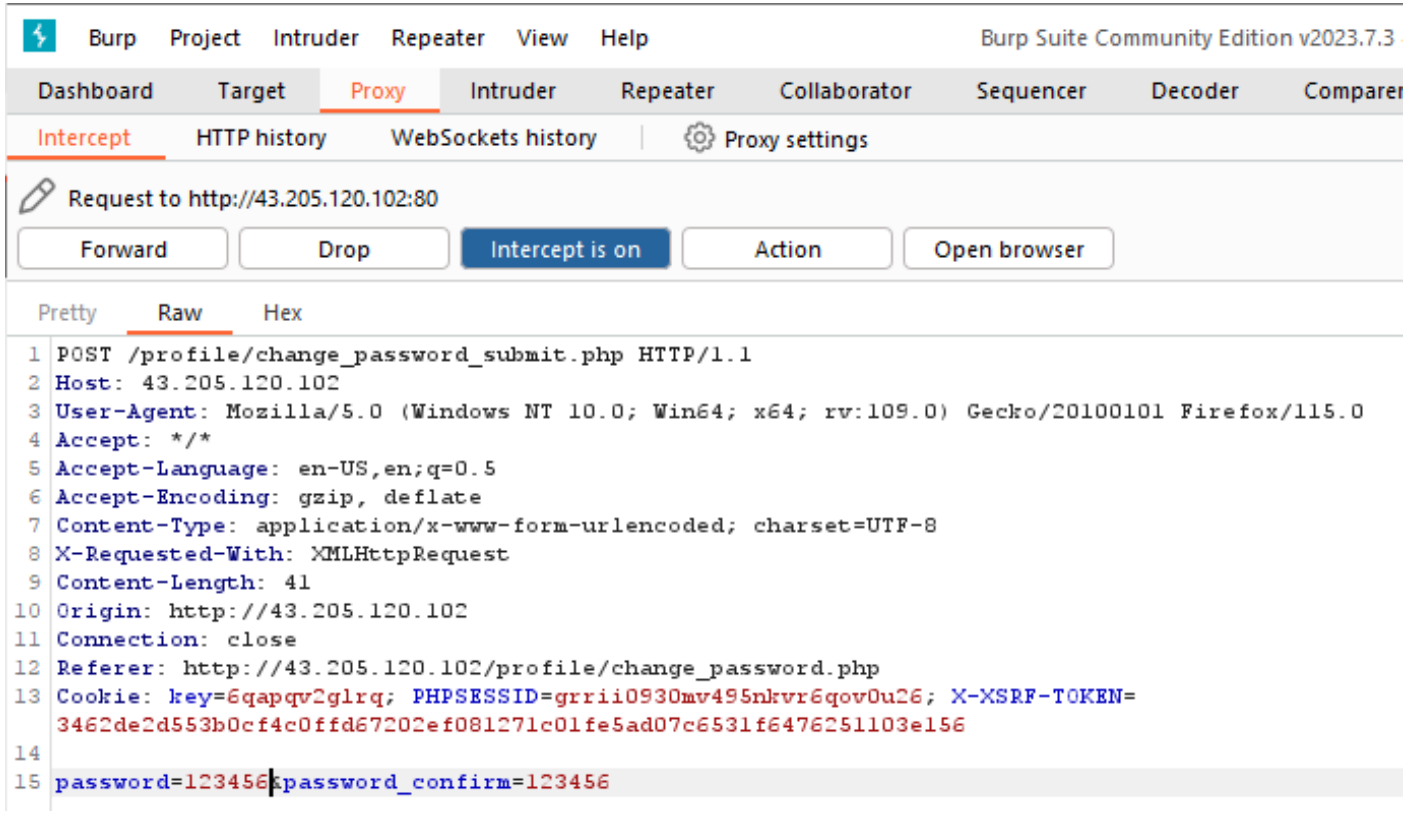
Observation

- see 7 digit pasword,but due to csrfI'll change the password at the moment he want to update.

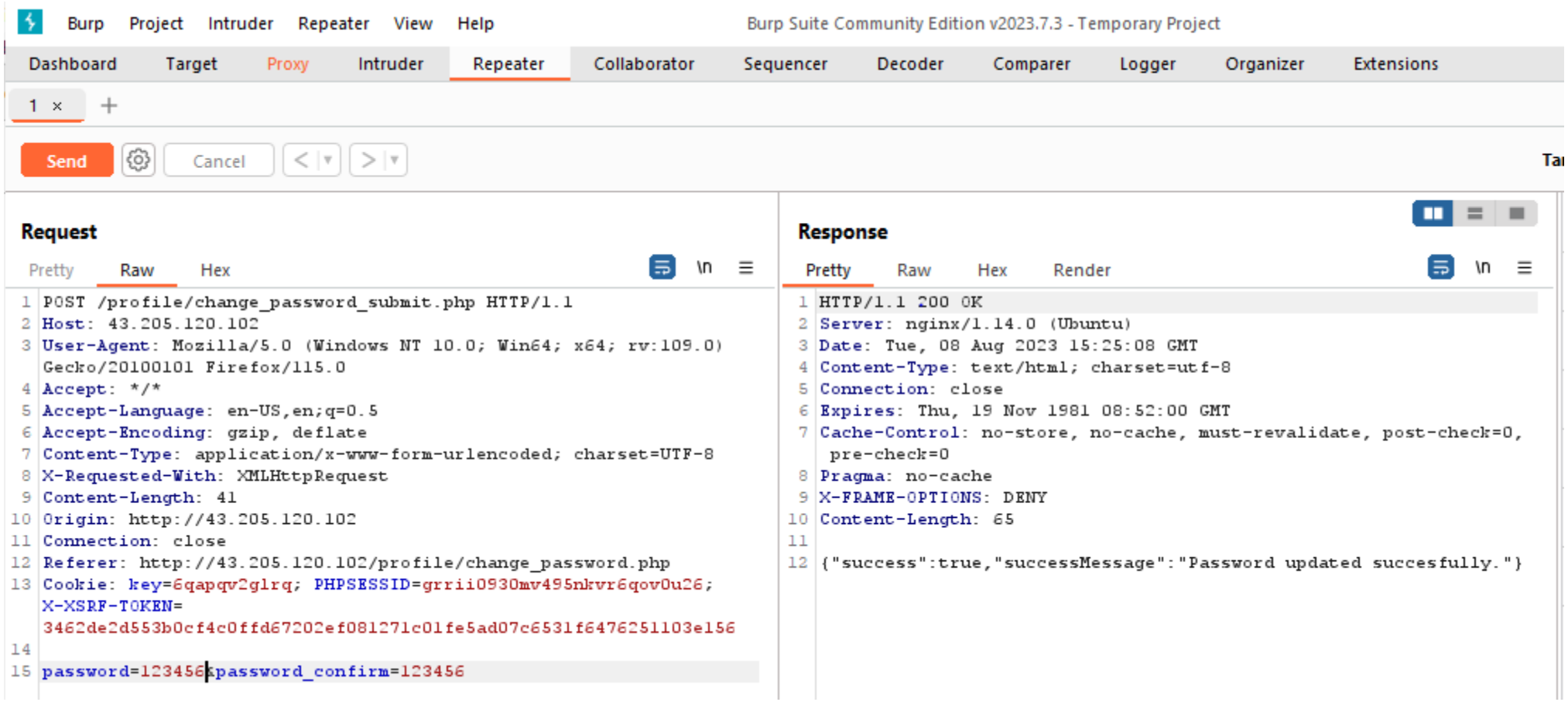


Observation

Here's the file I opened while chnagingpassword , when we click on send the password will change to 123456.



Proof Of Concept - POC



Business Impact - High

- Hacker can change the password of any user .
- Hacker can make user to do unwanted things
- It makes very bad impact of the website in the front of user

Recommendation

Take the following precautions:

- Implement an Anti-CSRF Token.
- Do not show the customers of the month on the login page.
- Use the Same Site Flag in Cookies.
- Check the source of request made.
- Take some extra keys or tokens from the user before processing an important request.
- Use 2 factor confirmations like otp, etc. for critical requests

References:

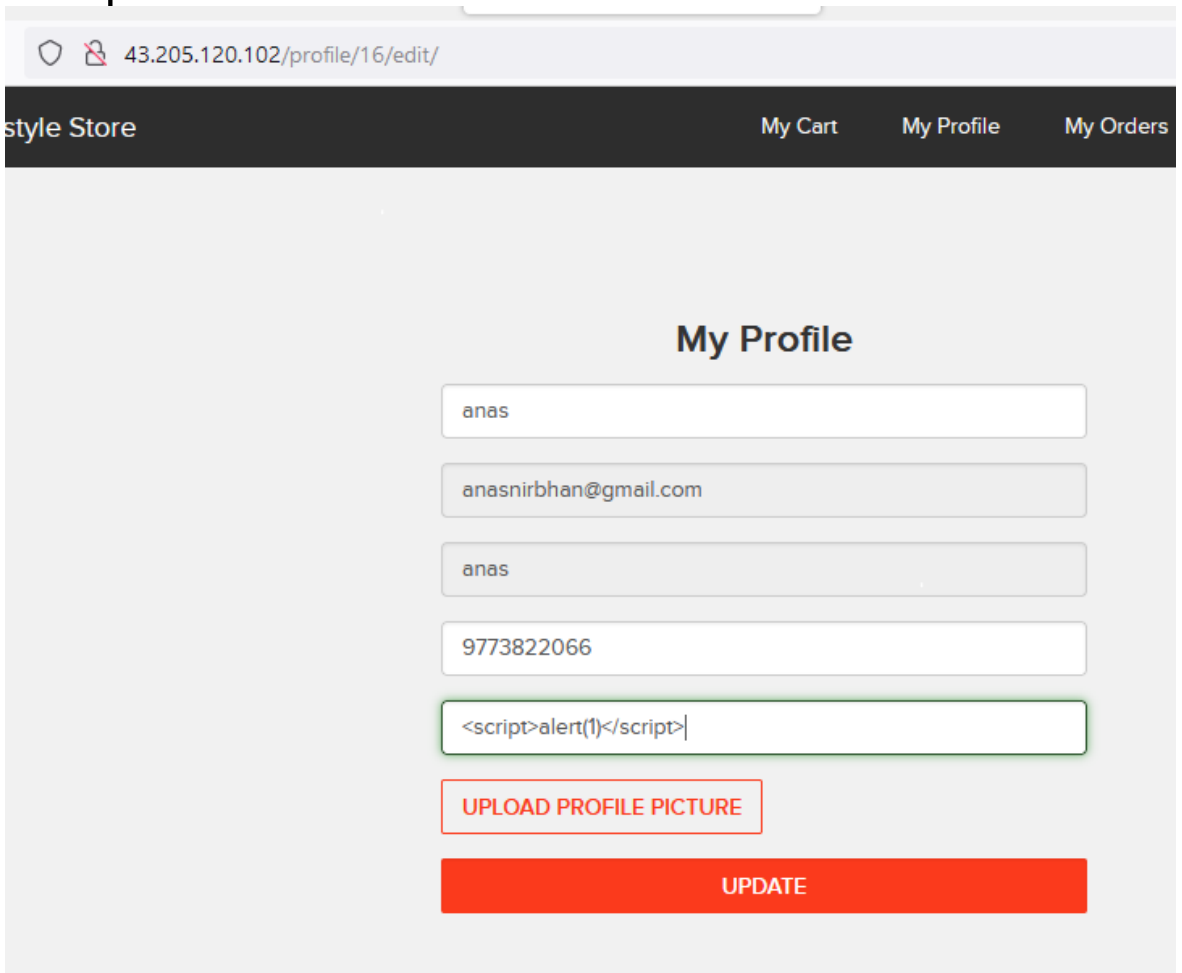
- https://www.netsparker.com/blog/web-security/csrf-cross-site-request-forgery/
- https://digitalguardian.com/blog/how-secure-personally-identifiable-information-against-loss-or-compromise

7. Reflected Cross Site Scripting

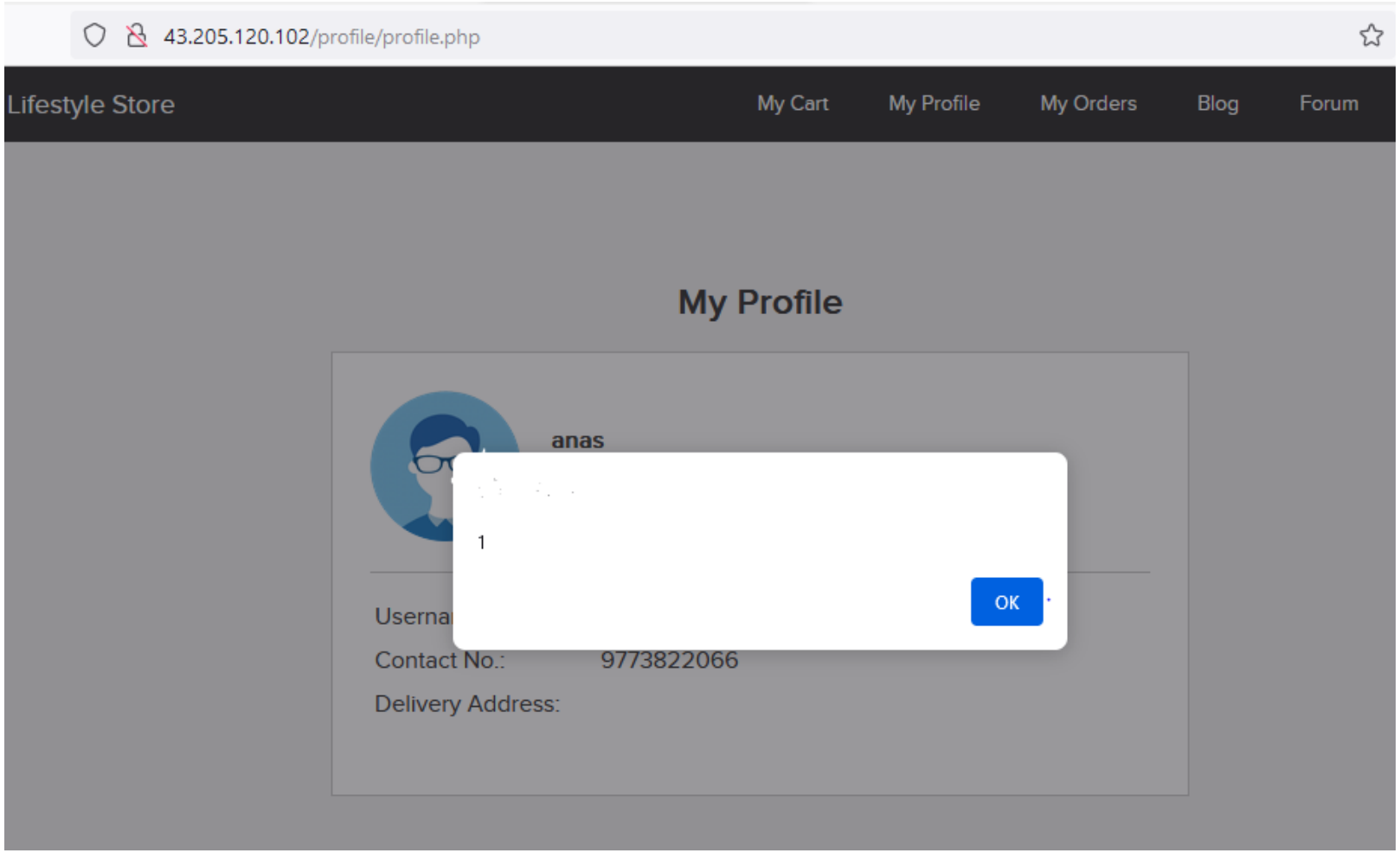
Reflected Cross Site Scripting (Severe)	Below mentioned parameters are vulnerable to reflected XSS **Affected URL :** •http://43.205.120.102/profile/16/edit/ **Affected Parameters :** •address(POST parameters) **Payload:** •<script>alert(1)</script>

Observation

- Open edit profile through URL and write the script on address bar



Proof Of Concept



Business Impact - High

- As attacker can inject arbitrary HTML CSS and JS via the URL, attacker can put any content on the page like phishing pages, install malware on victim’s device and even host explicit content that could compromise the reputation of the organization
- All attacker needs to do is send the link with the payload to the victim and victim would see hacker controlled content on the website. As the user trusts the website, he/she will trust the content.

Recommendation

Take the following precautions:

- Sanitize all user input and block characters you do not want
- Convert special HTML characters like ‘ “ < > into HTML entities " %22 < > before printing them on the website

References:

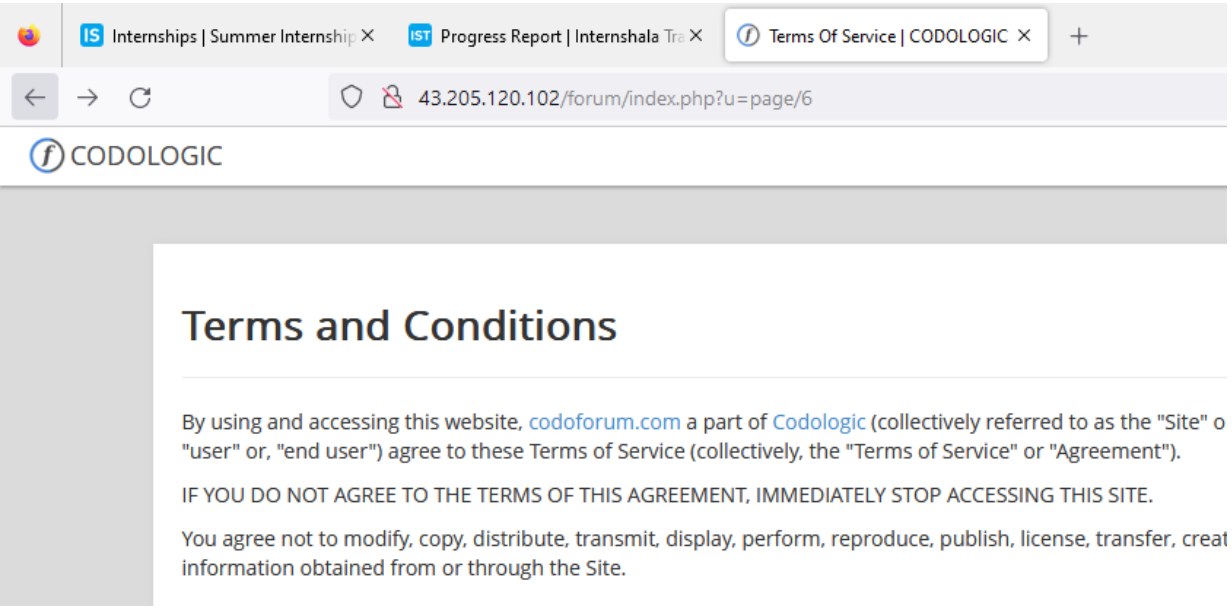
- *https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)*
- https://en.wikipedia.org/wiki/Cross-site_scripting
- https://www.w3schools.com/html/html_entities.asp

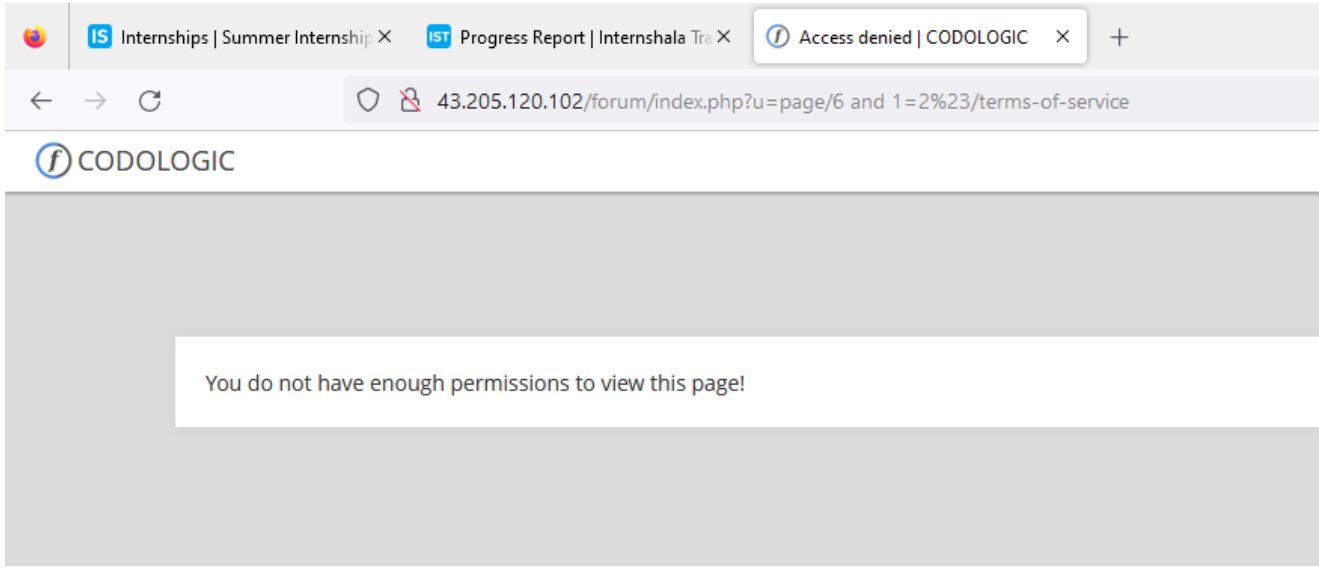
8. Components with known vulnerability

Components with known vulnerability (Severe)	•Server used is nginx/1.14.0 appears to be outdated (current is at least 1.17.3) It is known to have exploitable vulnerabilities. •WonderCMS •Codoforum (Powered by codologic)

Observation

Here we can see that the website is suffering from blind SQL injection





Proof Of Concept

Codologic Vulnerability,
It has multiple sql injection vulnerability,

Refer the link of exploit-db in reference section.

```
Proof of Concept:


        http://localhost/codoforum/index.php?u=/page/6 and
1=1%23/terms-of-service
        -> true (terms and services displayed)
        http://localhost/codoforum/index.php?u=/page/6 and
1-2%23/terms-of-service
        -> false ("You do not have enough permissions to view this page!")

Code:

        routes.php:593

            $pid = (int) $id;
            $user = \CODOF\User\User::get();

            $qry = 'SELECT title, content FROM ' . PREFIX . 'codo_pages p '
                    . ' LEFT JOIN ' . PREFIX . 'codo_page_roles r ON
r.pid=p.id '
                    . ' WHERE (r.rid IS NULL OR  (r.rid IS NOT NULL AND
r.rid IN (' . implode($user->rids) . ')))'
                    . ' AND p.id=' . $id;
```

Business Impact - High

Exploits of every vulnerability detected is regularly made public and hence outdated software can very easily be taken advantage of. If the attacker comes to know about this vulnerability, he/she may directly use the exploit to take down the entire system, which is a big risk.

Recommendation

- Upgrade to the latest version of Affected Software/theme/plugin/OS which means latest version.
- If upgrade is not possible for the time being, isolate the server from any other critical data and servers.

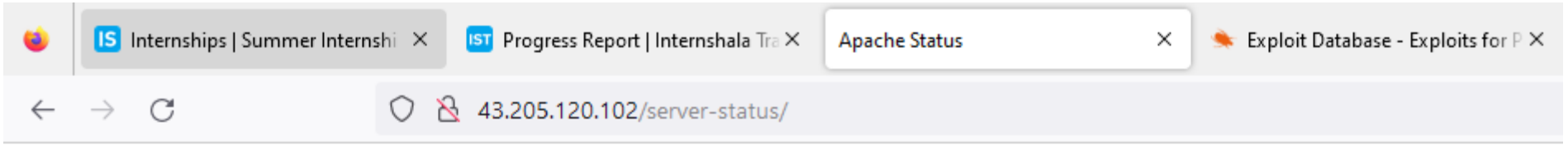
Reference

- https://usn.ubuntu.com/4099-1/(for ubuntu)
- https://www.exploit-db.com/exploits/37820
- https://securitywarrior9.blogspot.com/2018/01/vulnerability-in-wonder-cms-leading-to.html

9. Server Misconfiguration

Server Misconfiguration (Severe)	Below mentioned url that shows you some important server related information. Here the Server is not properly configured as these information is publicly available to everyone. This could work as a great asset for an attacker **URL** **http://43.205.120.102/server-status/**

Observation + Proof Of Concept POC



Apache Server Status for localhost (via 127.0.0.1)

Server Version: Apache/2.4.18 (Ubuntu)
Server MPM: event
Server Built: 2018-06-07T19:43:03

Current Time: Monday, 05-Nov-2018 14:46:35 IST
Restart Time: Monday, 05-Nov-2018 09:14:47 IST
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 5 hours 31 minutes 47 seconds
Server load: 1.34 1.26 1.06
Total accesses: 35 - Total Traffic: 97 kB
CPU Usage: u8.1 s11.23 cu0 cs0 - .0971% CPU load
.00176 requests/sec - 4 B/second - 2837 B/request
1 requests currently being processed, 49 idle workers

PID	Connections		Threads		Async connections		
	total	accepting	busy	idle	writing	keep-alive	closing
1709	0	yes	0	25	0	0	0
1710	1	yes	1	24	0	1	0
Sum	1		1	49	0	1	0

Recommendation

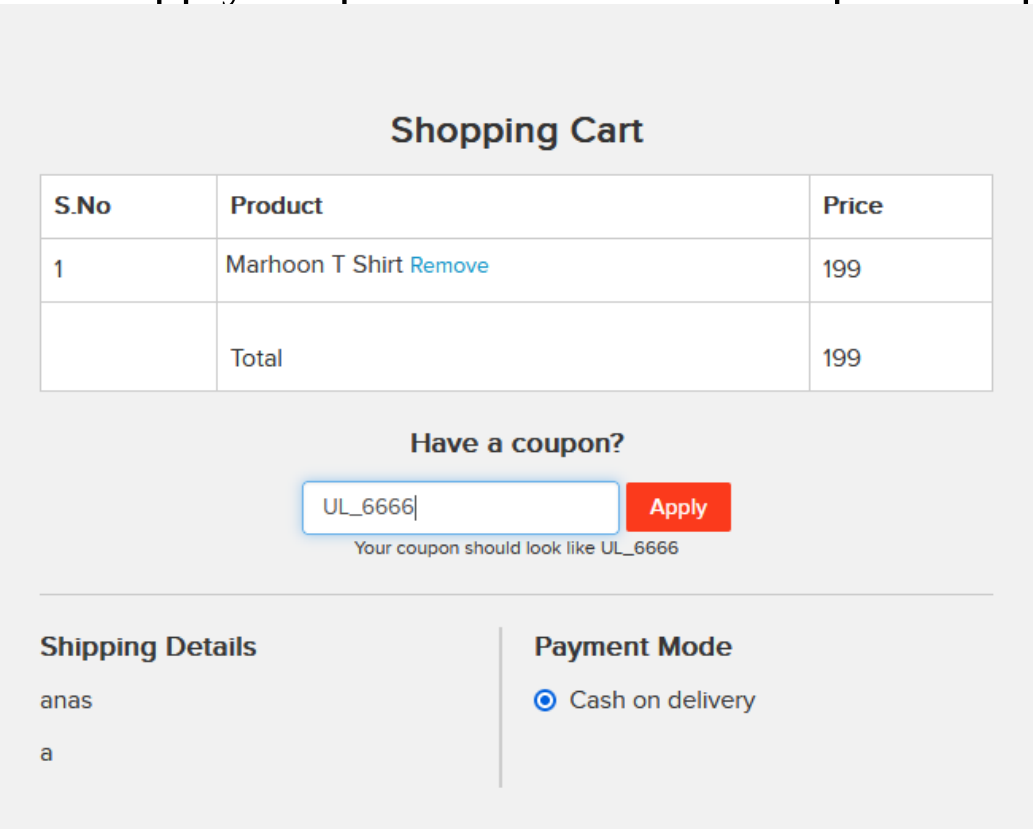
- Keep the software up to date
- Develop strong app architecture and encrypt data which has sensitive information
- Make sure that the security settings in the framework and libraries are set to secured values
- Perform regular audits and run tools to identify the holes in the System

10. Brute Force – Coupon Code

Brute Force (Severe)	Below mentioned URL is vulnerable to brute forcing and can be exploited for discounts. Affected URL : • http://15.207.106.113/cart/apply_coupon.php

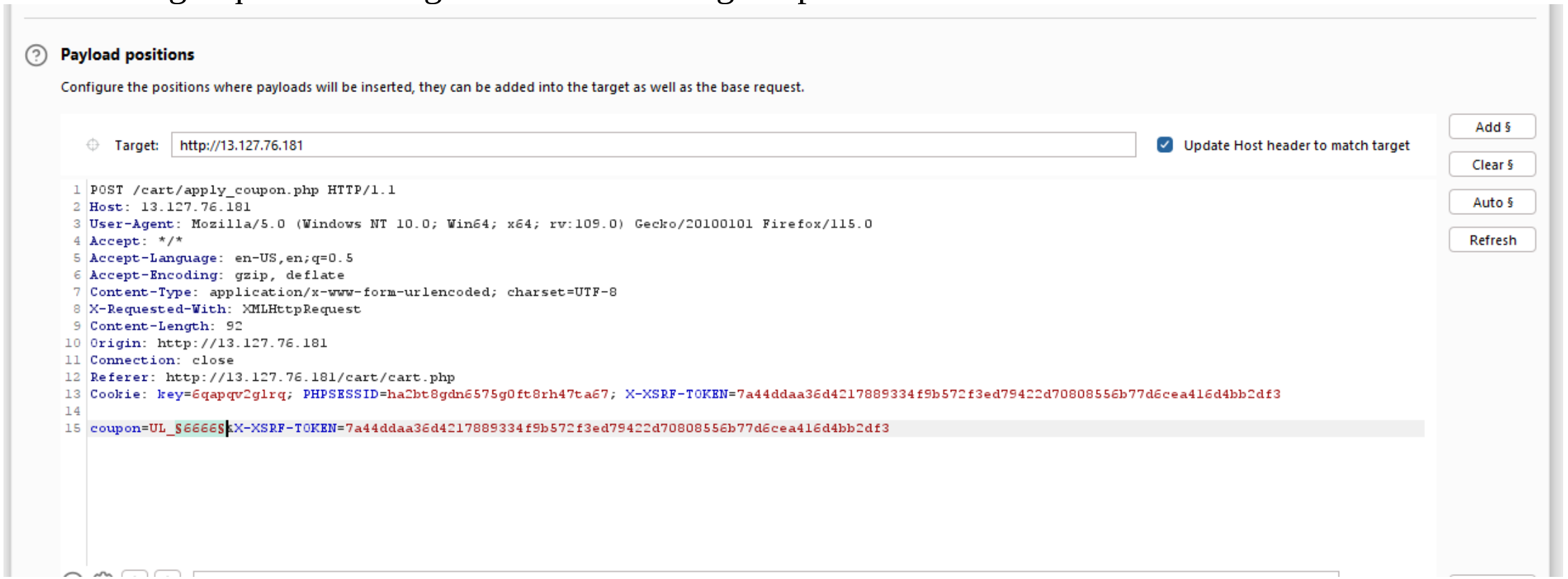
Observation

- Upon adding items to the cart, you will end up in a screen like this, where we see the apply coupon section and an example.
- Type in UL_6666 in the apply coupon section and intercept the request using Burp Suite



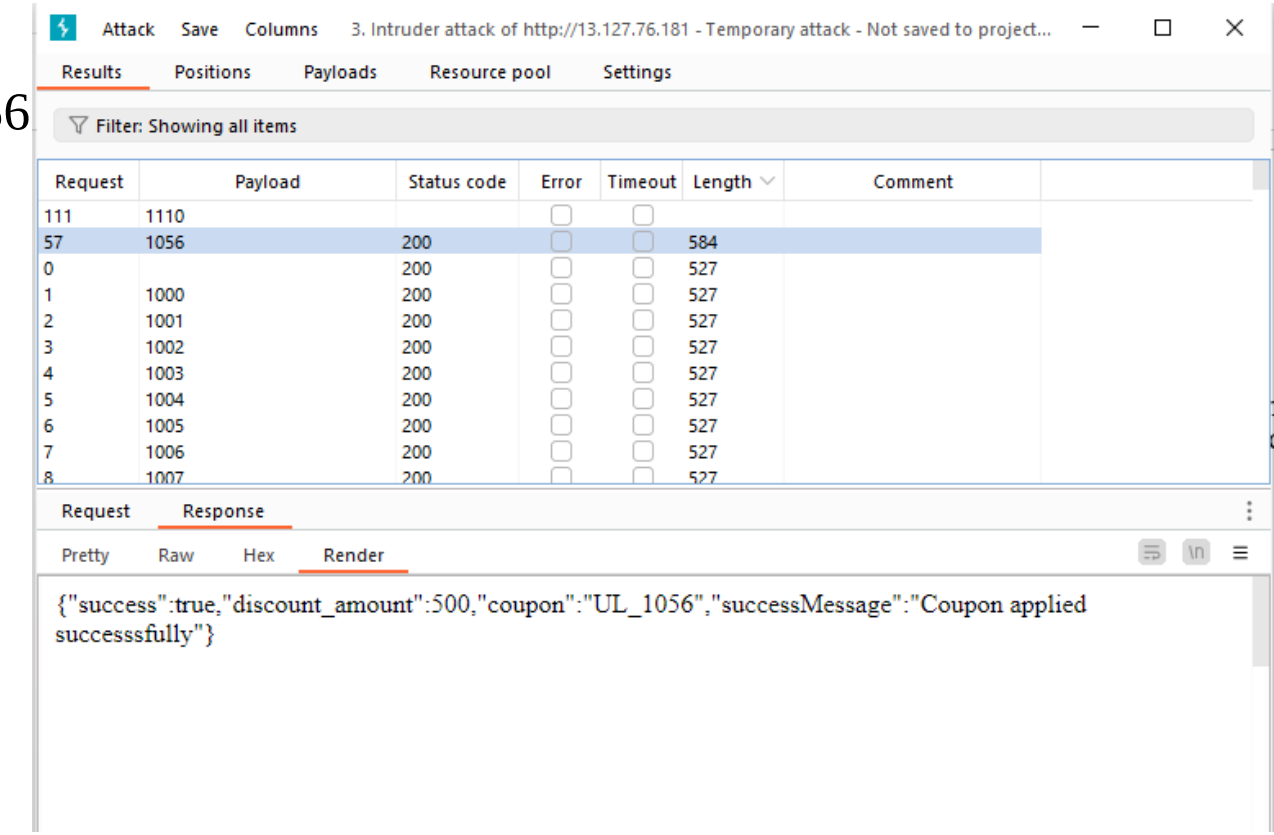
Observation

Following request will be generated containing coupon code



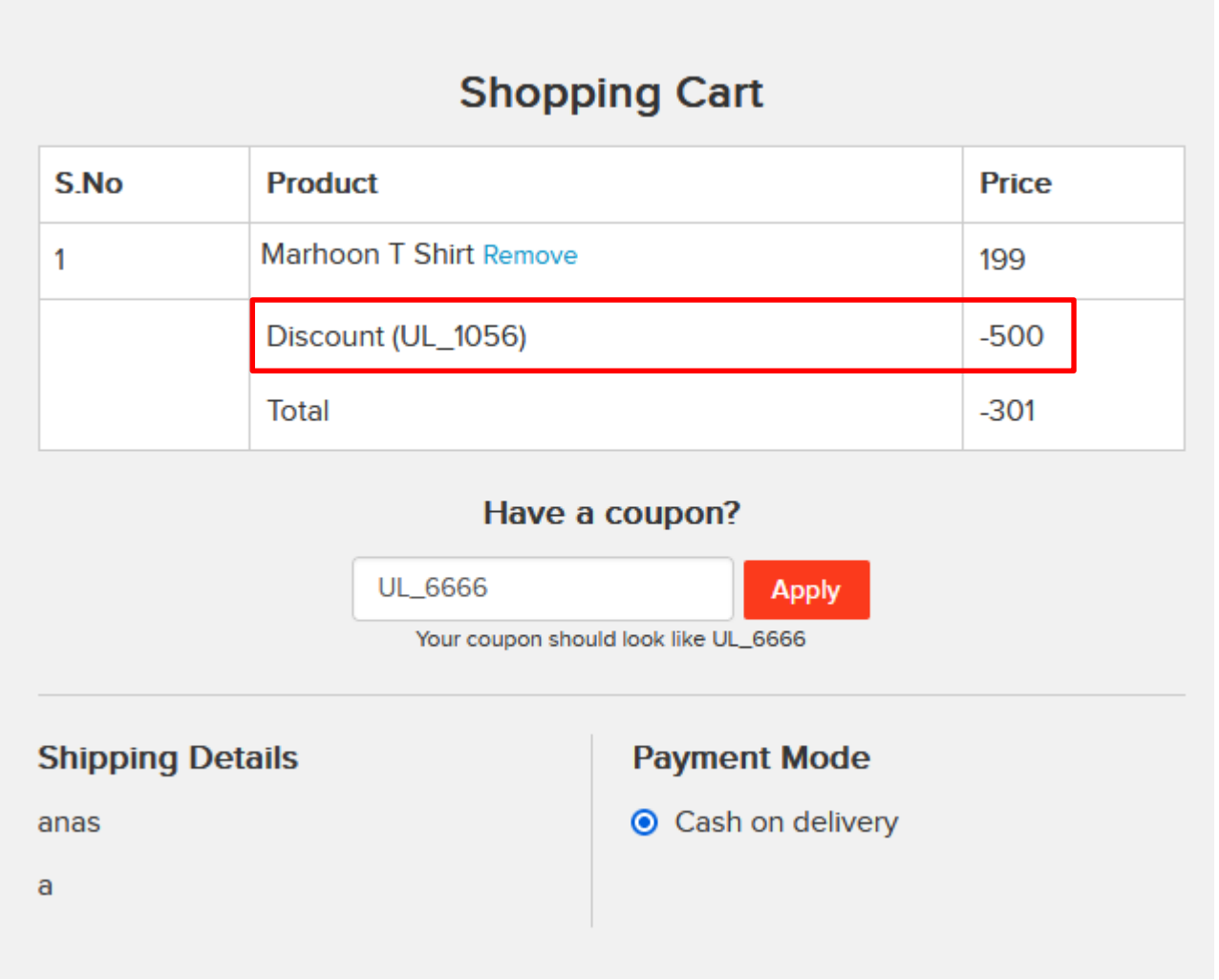
Observation

- We shoot the request with all possible combinations of 4 Digit numbers and upon a successful hit, we get a response containing the valid coupon code. We can use this code to get the discount.
- Valid coupon code for this website is UL_1056



Proof Of Concept

Coupon Applied Successfully.



Business Impact - Severe

An attacker or a malicious user can easily order the items on extreme discounts which in turn will cause huge financial loss to the company.

The loss may not be noticeable in short term but it will surely affect in long term.

Recommendation

- Coupon codes should have limited number of uses and should be regenerated after sometime.
- Coupon code should be random alpha-numeric characters

References

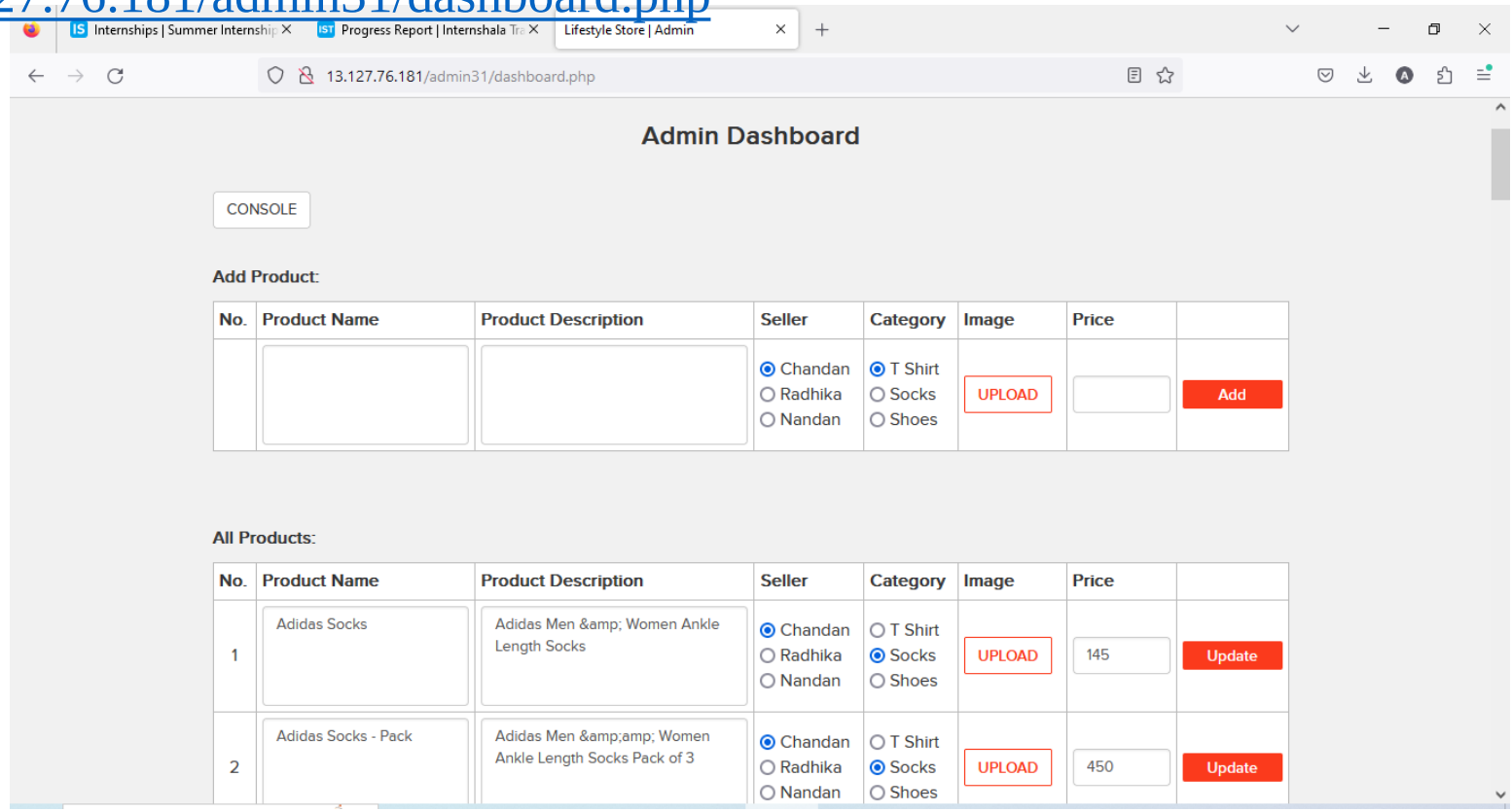
- https://www.digitalcommerce360.com/2017/03/17/prevent-fraud-brute-force-online-coupon-gift card-attacks/
- https://www.couponxoo.com/brute-force-attack-coupon-code

11. Forced Browsing

Forced Browsing (Severe)	Below mentioned URLs is vulnerable to forced browsing. Affected URL : • http://13.127.76.181/ Forced URLs : • http://13.127.76.181/admin31/dashboard.php • http://13.127.76.181/admin31/console.php

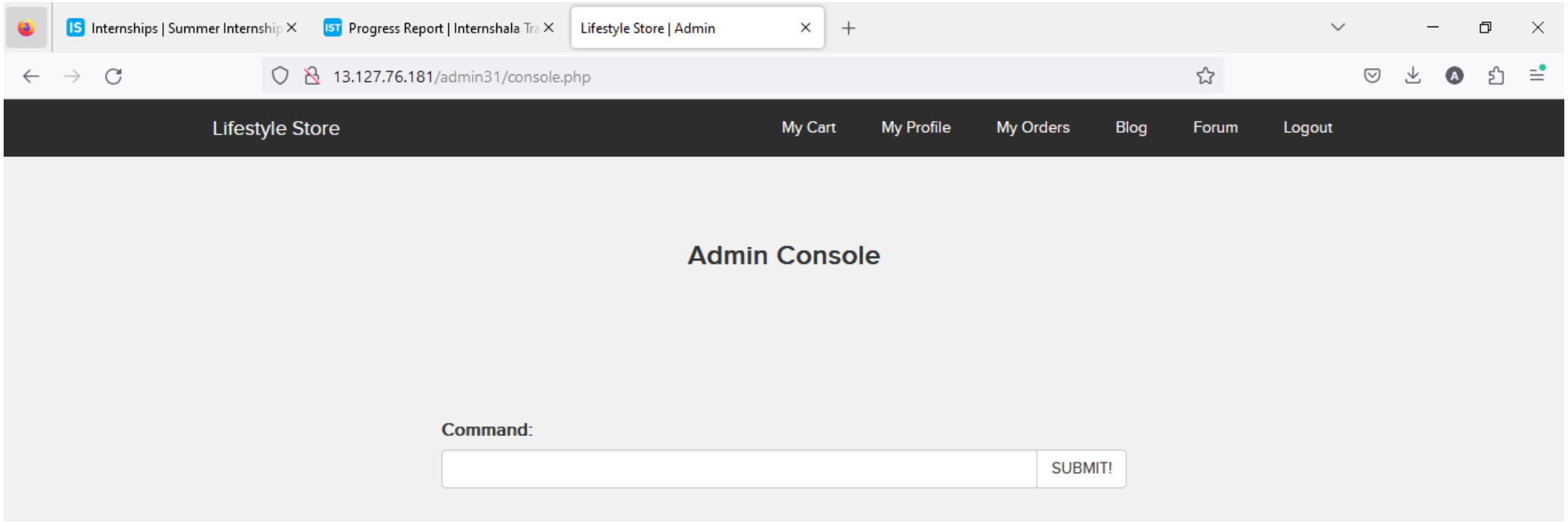
Observation and POC

- As a customer, Login to your account.
- Now, forcefully type in the url for going to the admin dashboard http://13.127.76.181/admin31/dashboard.php



Proof Of Concept – Admin Console

Here is the access to the admin console just by entering its complete url.



Business Impact -Severe

- Attacker can have all the admin privileges.
- He can edit all the items.
- He can execute any harmful command through console.

Recommendation

- Server side security checks should be performed perfectly.
- Make the admin page url complicated so that it couldn’t be guessed.

References

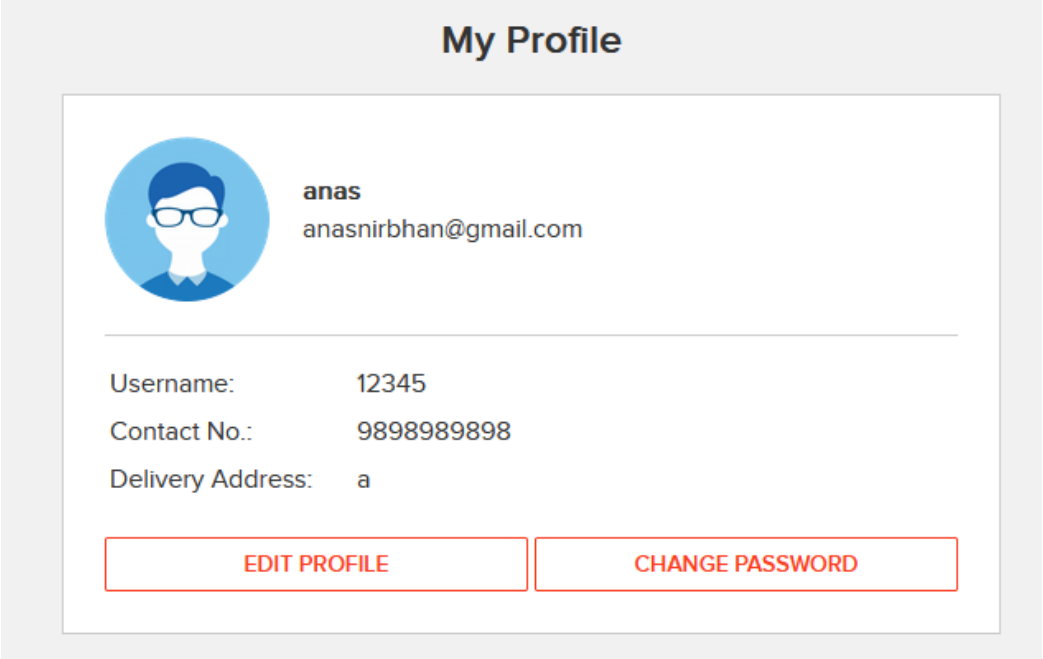
- https://owasp.org/www-community/attacks/Forced_browsing
- https://campus.barracuda.com/product/webapplicationfirewall/doc/42049348/forced-browsing attack/

12. Client Side Filter Bypass

Client Side Filter Bypass (Moderate)	Below mentioned URL is vulnerable to client side filter bypass. Affected URL : • http://3.6.40.63/profile/16/edit/

Observation

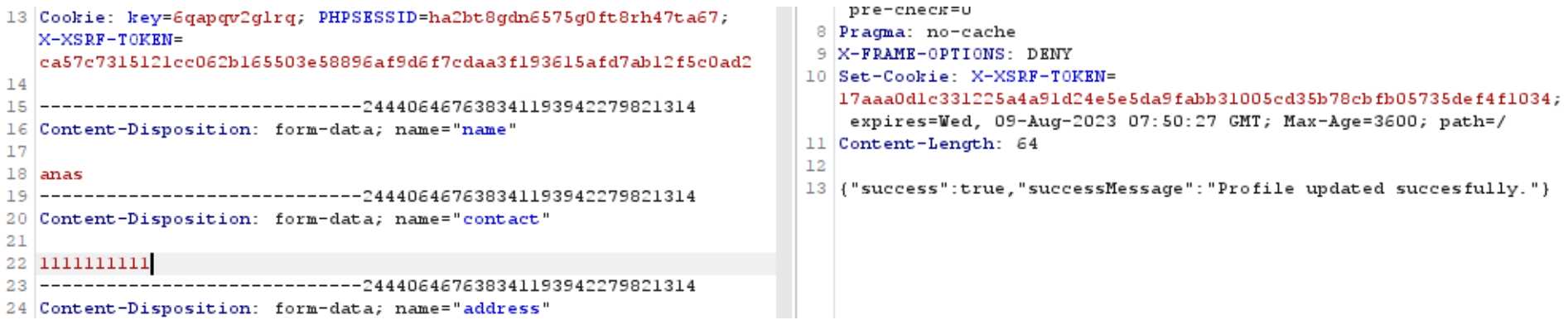
- S1 - Login to your account and go to My Profile section.
- S2 - Now, click on edit profile button, update any of your details, here I have chosen the phone number to proceed.
- S3 - I updated my phone number from 977382xxxx to 9898989898.
- S4 - Now, again click on UPDATE button and intercept the request with Burp Suite.



Observation

The client side filter doesn't allow to enter such (111111111111) as phone number, showing an error message that please enter valid number.

- Now, send the request to the Repeater and edit the phone number.
- I changed it from 9898989898 to 1111111111 and hit Send.

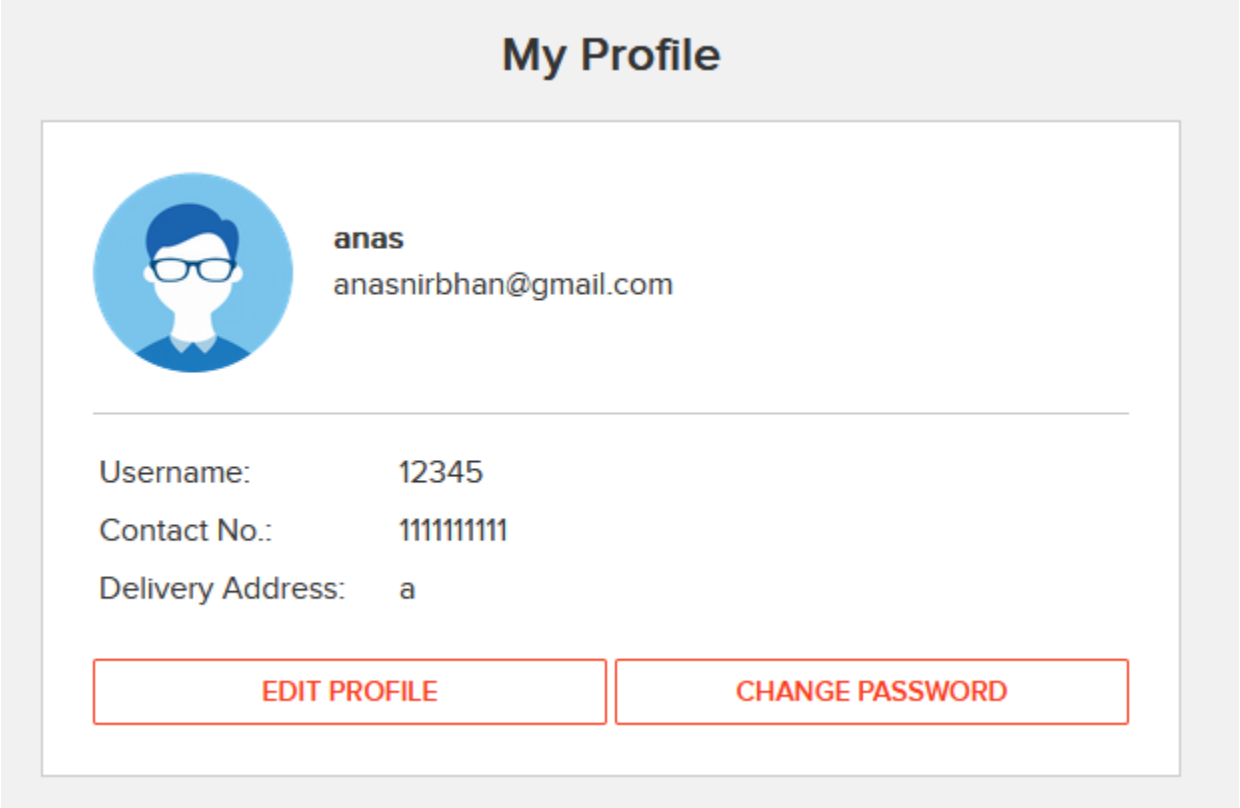


```
13 Cookie: key=6qapqv2glrq; PHPSESSID=ha2bt8gdn6575g0ft8rh47ta67;
   X-XSRF-TOKEN=
   ca57c7315121cc062b165503e58896af9d6f7cdaa3f193615afd7ab12f5c0ad2
14
15 -----------------------------244406467638341193942279821314
16 Content-Disposition: form-data; name="name"
17
18 anas
19 -----------------------------244406467638341193942279821314
20 Content-Disposition: form-data; name="contact"
21
22 1111111111
23 -----------------------------244406467638341193942279821314
24 Content-Disposition: form-data; name="address"
```

```
   pre-check=0
 8 Pragma: no-cache
 9 X-FRAME-OPTIONS: DENY
10 Set-Cookie: X-XSRF-TOKEN=
   17aaa0d1c331225a4a91d24e5e5da9fabb31005cd35b78cbfb05735def4f1034;
   expires=Wed, 09-Aug-2023 07:50:27 GMT; Max-Age=3600; path=/
11 Content-Length: 64
12
13 {"success":true,"successMessage":"Profile updated succesfully."}
```

Proof Of Concept POC

Profile Updated Successfully.



Business Impact - High

- This would only trouble the users who in turn might give negative feedback on your website.
- This could cause the severe effect on the organization’s reputation

Recommendation

Take the following precautions:

- Implement all critical checks on server side code only.
- Client-side checks must be treated as decorative only.
- All business logic must be implemented and checked on the server code.
- This includes user input, the flow of applications and even the URL/Modules a user is supposed to access or not

References

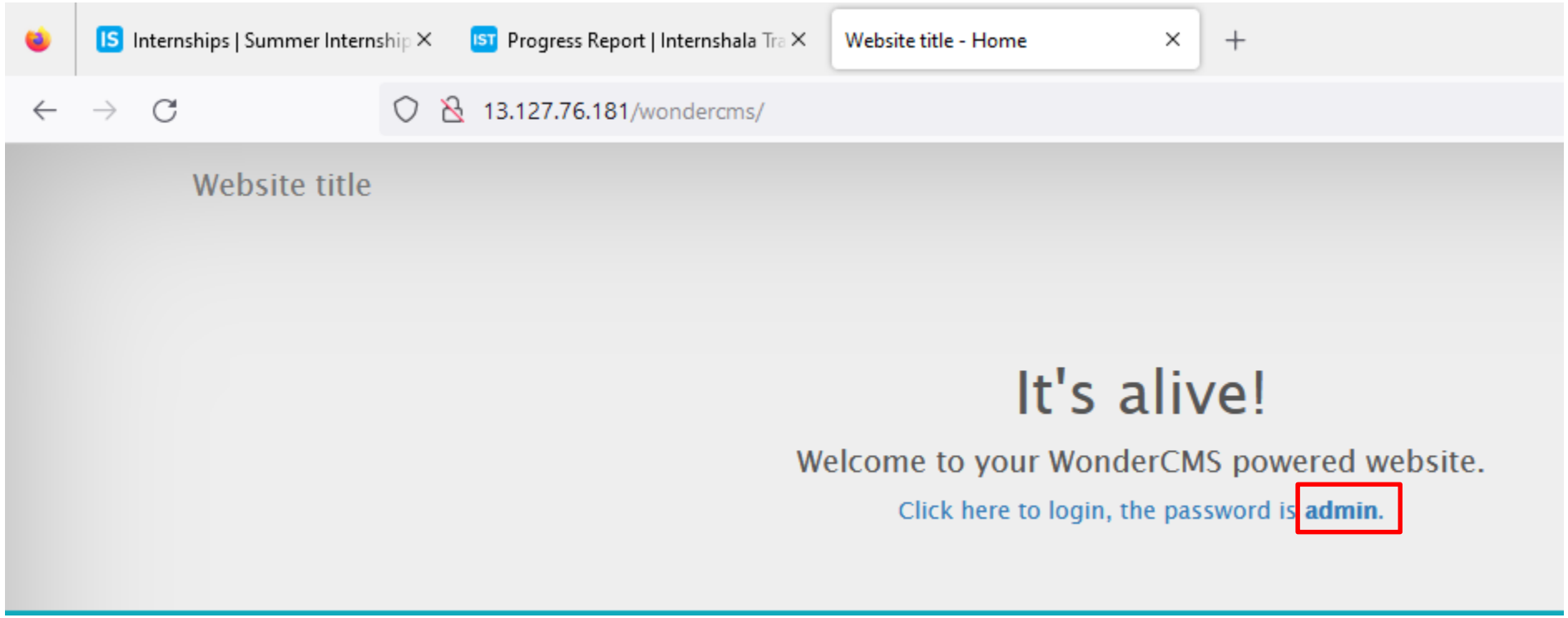
- https://portswigger.net/support/using-burp-to-bypass-client-side-javascript-validation
- https://www.slideshare.net/SamBowne/cnit-129s-ch-5-bypassing-clientside-controls

13. Default/Common Password

Default/Common Password (Severe)	Below mentioned url has Default and very common password. The admin password is just directly displayed on the webpage which is a very severe mistake and cause a lot of damage. **Affected URL :** • http://13.127.76.181/wondercms/

Observation

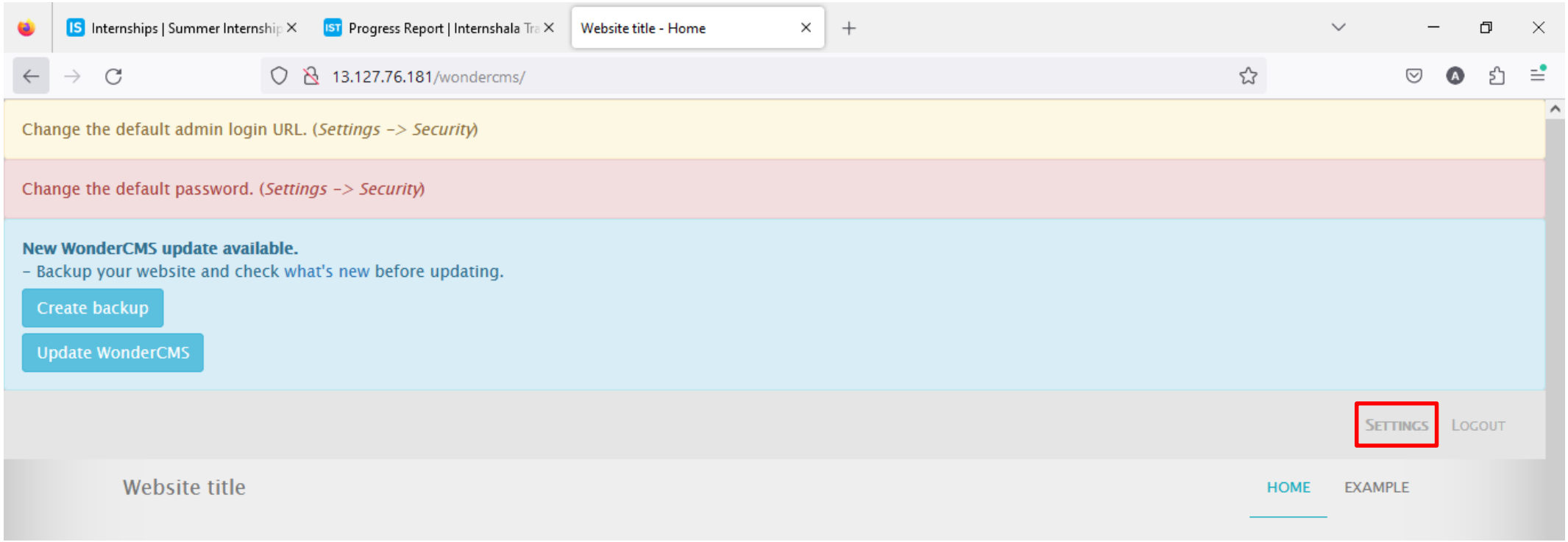
Here you can clearly see the admin password is displayed openly on the web page.



Proof Of Concept - POC

Here you can clearly see that using this password one can get access to the admin panel of the website.

For an Attacker this is not less than a treasure.



Business Impact - High

Easy, default and common passwords make it easy for attackers to gain access to their accounts illegal use of them and can harm the website to any extent after getting logged into privileged accounts.

Recommendation

- There should be password strength check at every creation of an account.
- There must be a minimum of 8 characters long password with a mixture of numbers, alphanumerics, special characters ,etc.
- There should be no repetition of password ,neither on change nor reset.
- The password should not be stored on the web, rather should be hashed and stored

References

- https://www.acunetix.com/blog/articles/weak-password-vulnerability-common-think/
- https://www.owasp.org/index.php/Testing_for_Weak_password_policy_(OTG-AUTHN-007)

14. IDOR-Unauthorized access to user details

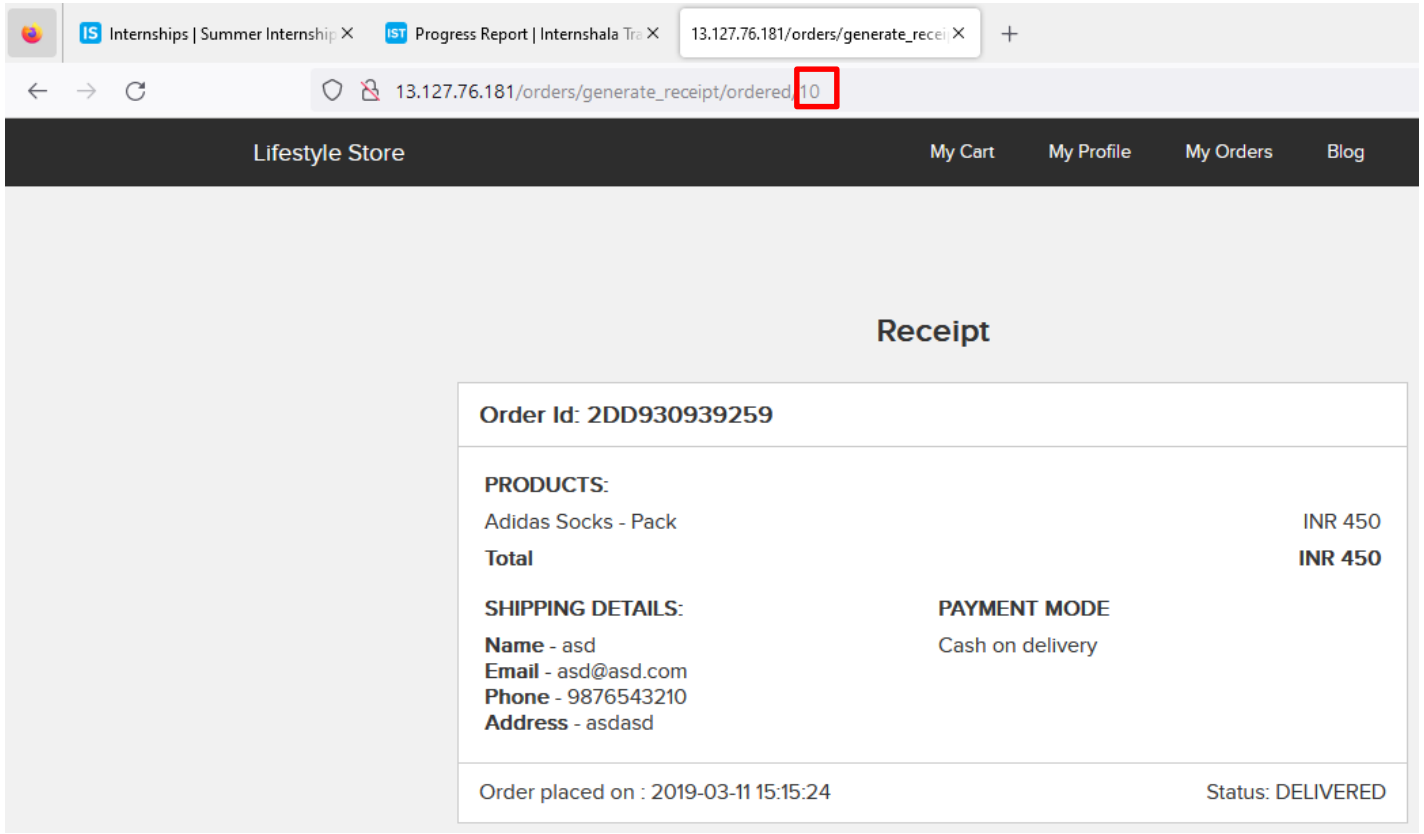
IDOR-Unauthorized access to user details (Severe)	Below mentioned url will have vulnerability through which anyone can see the details of another user **URL** http://13.127.76.181/orders/generate_receipt/ordered/11 Affected parameter Ordered/**11** **Payload** http://13.233.173.221/generate_receipt/ordered/10

14. IDOR-Unauthorized access to user details

IDOR-Unauthorized access to user details (Severe)	Below mentioned url have vulnerability through which anyone can see the details of another user. You just have to change the numeric value given in the url's. They can be seen as customer id. **URL'S effected:-** **http://13.127.76.181/orders/orders.php?customer=14** **http://13.127.76.181/profile/14/edit/** **http://13.127.76.181/forum/index.php?u=/user/profile/4** All the URLS mentioned above are those through which I accessed the details of a user having customerid = 14 and in forum profile number 4.

Observation and POC

Originally my URL was http://13.127.76.181/orders/generate_receipt/ordered/11 but, by inserting a payload i.e (making 11 as 10) I was able to access the receipt details of other user.



Proof Of Concept - POC

A very detailed POC can be found in Recordings > IDOR > IDOR_Recording

Business Impact - High

A malicious hacker can read bill information and account details of any user just by knowing the customer id and User ID. This discloses critical billing information of users including:

- **Mobile Number**
- **Bill Number**
- **Billing Period**
- **Total number of orders ordered by customer**
- **Bill Amount and Breakdown**
- **Phone no. and email address**
- **Address**

Business Impact - High

This can be used by malicious hackers to carry out targeted phishing attacks on the users and the information can also be sold to competitors/ black market . More over, as there is no rate limiting checks, attacker can bruteforce the user_id for all possible values and get bill information of each and every user of the organization resulting is a massive information leakage.

Recommendation

The following precautions are recommended to be undertaken :

- Implement proper authentication and authorization checks to make sure that the user has permission to the data he/she is requesting
- Use proper rate limiting checks on the number of request comes from a single user in a small amount of time
- Make sure each user can only see his/her data only

References

- https://www.owasp.org/index.php/Insecure_Configuration_Management
- https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References

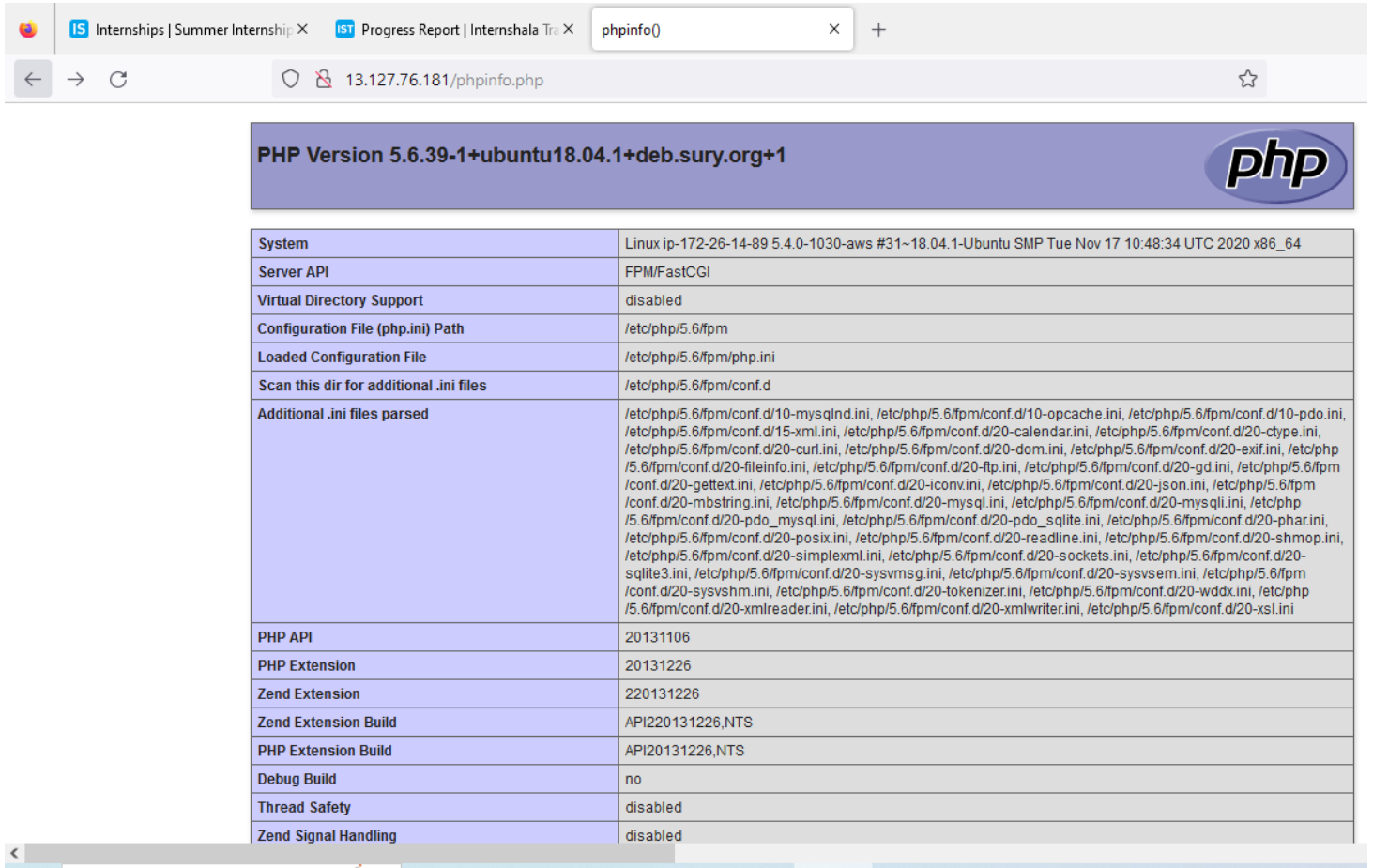
15. Directory Listing

Directory Listing (Moderate)	Below mentioned are some urls that are disclosing various server information that should not be known or easily available for a normal user. Affected URL : • http://13.127.76.181/phpinfo.php • http://13.127.76.181/robots.txt • http://13.127.76.181/composer.lock • http://13.127.76.181/composer.json • http://13.127.76.181/userlist.txt

Observation

Used this URL and found out different information and data related to the website server and other resources.

One of the output Image :



System	Linux ip-172-26-14-89 5.4.0-1030-aws #31~18.04.1-Ubuntu SMP Tue Nov 17 10:48:34 UTC 2020 x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/5.6/fpm
Loaded Configuration File	/etc/php/5.6/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/5.6/fpm/conf.d
Additional .ini files parsed	/etc/php/5.6/fpm/conf.d/10-mysqlnd.ini, /etc/php/5.6/fpm/conf.d/10-opcache.ini, /etc/php/5.6/fpm/conf.d/10-pdo.ini, /etc/php/5.6/fpm/conf.d/15-xml.ini, /etc/php/5.6/fpm/conf.d/20-calendar.ini, /etc/php/5.6/fpm/conf.d/20-ctype.ini, /etc/php/5.6/fpm/conf.d/20-curl.ini, /etc/php/5.6/fpm/conf.d/20-dom.ini, /etc/php/5.6/fpm/conf.d/20-exif.ini, /etc/php/5.6/fpm/conf.d/20-fileinfo.ini, /etc/php/5.6/fpm/conf.d/20-ftp.ini, /etc/php/5.6/fpm/conf.d/20-gd.ini, /etc/php/5.6/fpm/conf.d/20-gettext.ini, /etc/php/5.6/fpm/conf.d/20-iconv.ini, /etc/php/5.6/fpm/conf.d/20-json.ini, /etc/php/5.6/fpm/conf.d/20-mbstring.ini, /etc/php/5.6/fpm/conf.d/20-mysql.ini, /etc/php/5.6/fpm/conf.d/20-mysqli.ini, /etc/php/5.6/fpm/conf.d/20-pdo_mysql.ini, /etc/php/5.6/fpm/conf.d/20-pdo_sqlite.ini, /etc/php/5.6/fpm/conf.d/20-phar.ini, /etc/php/5.6/fpm/conf.d/20-posix.ini, /etc/php/5.6/fpm/conf.d/20-readline.ini, /etc/php/5.6/fpm/conf.d/20-shmop.ini, /etc/php/5.6/fpm/conf.d/20-simplexml.ini, /etc/php/5.6/fpm/conf.d/20-sockets.ini, /etc/php/5.6/fpm/conf.d/20-sqlite3.ini, /etc/php/5.6/fpm/conf.d/20-sysvmsg.ini, /etc/php/5.6/fpm/conf.d/20-sysvsem.ini, /etc/php/5.6/fpm/conf.d/20-sysvshm.ini, /etc/php/5.6/fpm/conf.d/20-tokenizer.ini, /etc/php/5.6/fpm/conf.d/20-wddx.ini, /etc/php/5.6/fpm/conf.d/20-xmlreader.ini, /etc/php/5.6/fpm/conf.d/20-xmlwriter.ini, /etc/php/5.6/fpm/conf.d/20-xsl.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled

Proof Of Concept - POC

For POC please visit the Directory Listing folder inside Screenshots.

Business Impact - Moderate

Although this vulnerability does not have a direct impact to users or the server, though it can aid the attacker with information about the server and the users. Information Disclosure due to default pages are not exploitable in most cases, but are considered as web application security issues because they allows malicious hackers to gather relevant information which can be used later in the attack lifecycle, in order to achieve more than they could if they didn’t get access to such information.

Recommendation

- Disable all default pages
- Enable multiple security checks

References

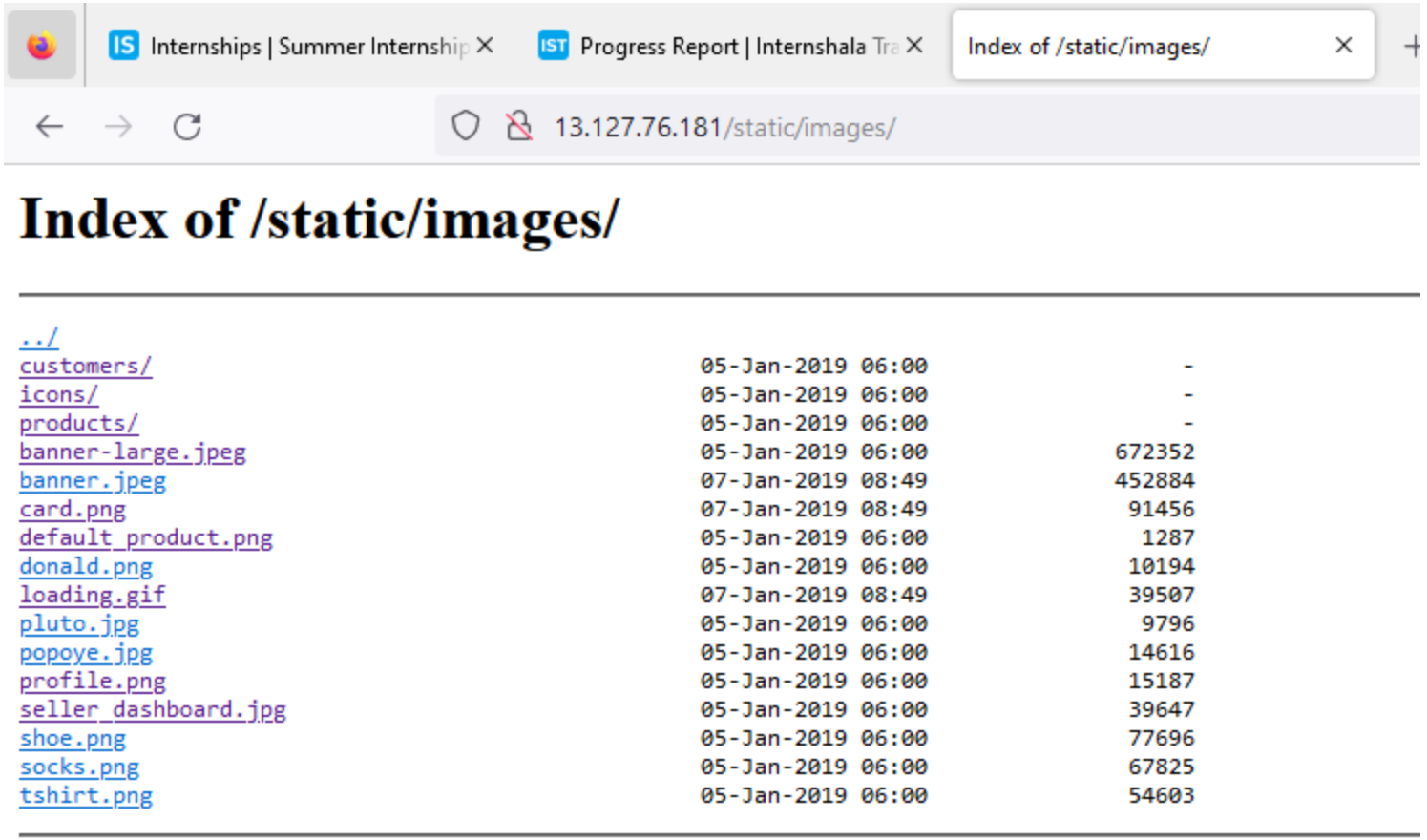
- https://www.netsparker.com/blog/web-security/information-disclosure-issues-attacks/
- https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/information-disclosure-phpinfo/

16. Personal Information Leakage

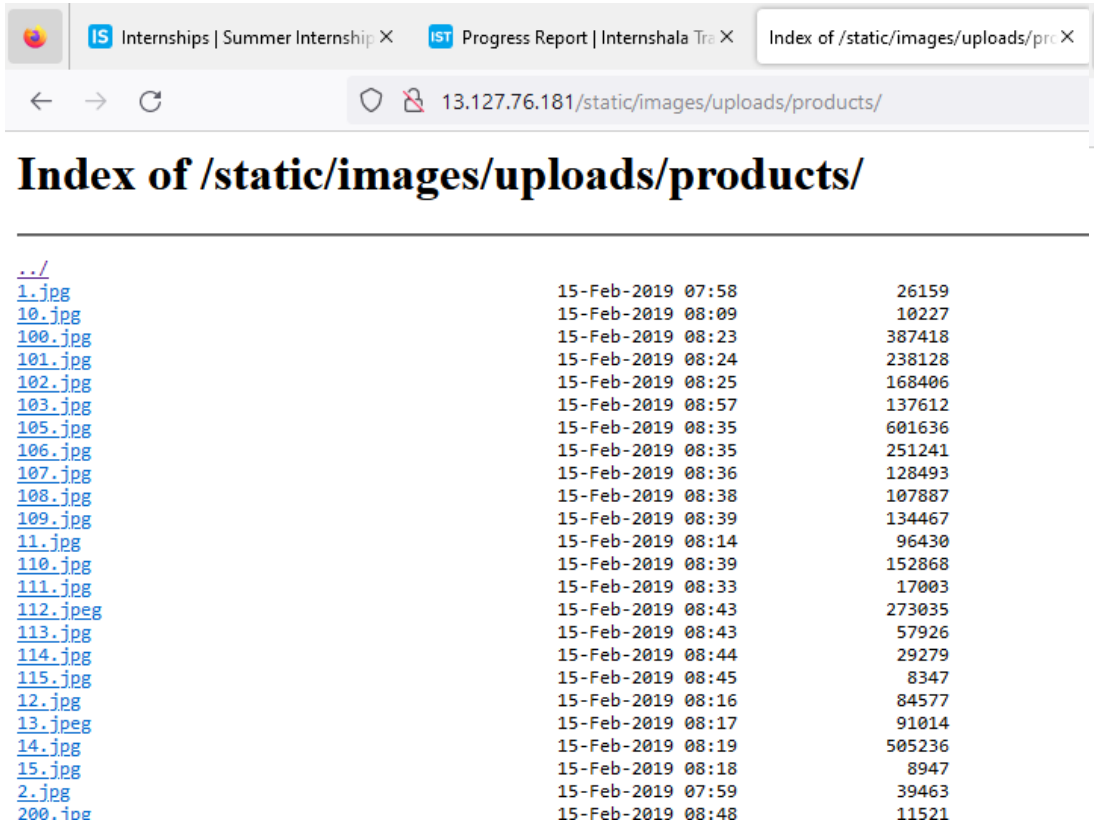
Personal Information Leakage (Low)	Below mentioned urls disclose personal information **Affected URL :** • http://13.127.76.181/static/images/ • http://13.127.76.181/static/images/customers/default.png • http://13.127.76.181/products/details.php?p_id=2

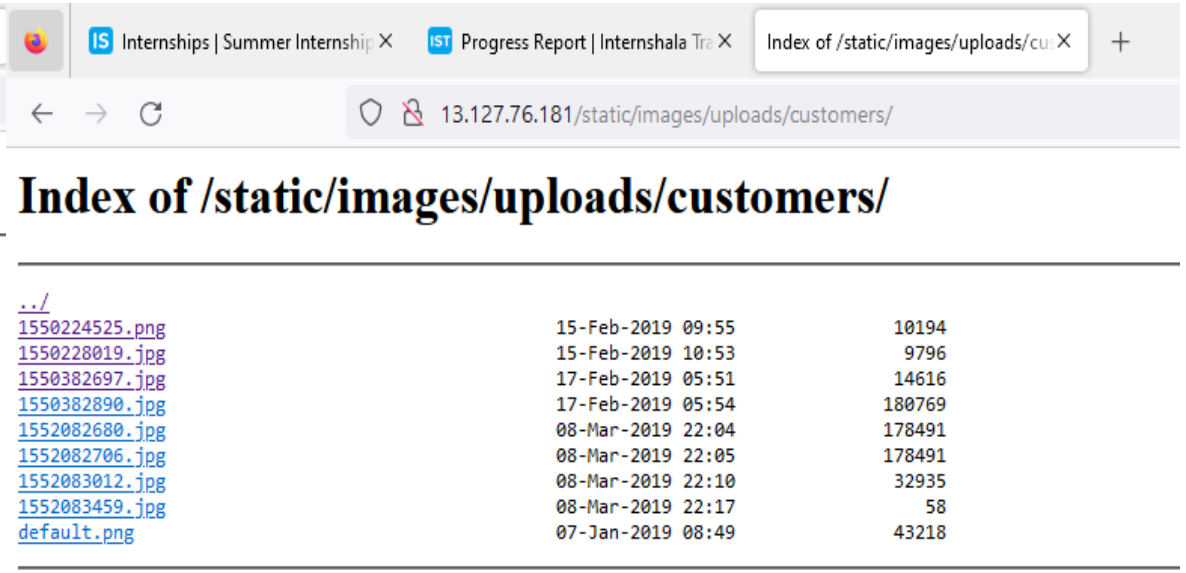
Observation

By navigating to the URL mentioned one can basically see the data stack / folder where all the images of different sections are stored.



Proof Of Concept - POC





Business Impact - Moderate

Although this vulnerability does not have a direct impact to users or the server, though it can help the attacker in mapping the personal information of any account and plan further attacks on any specific account.

Recommendation

- You can apply encryption to the personal data
- You can add authenticity and authorization to access the other data

References

- https://cipher.com/blog/25-tips-for-protecting-pii-and-sensitive-data/
- https://digitalguardian.com/blog/how-secure-personally-identifiable-information-against-loss-or-compromise

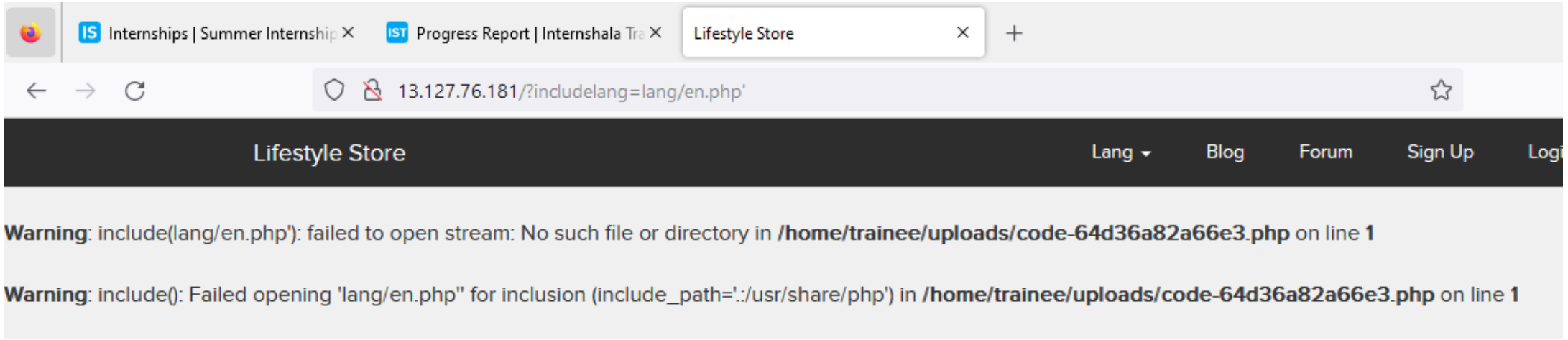
17. Default Messages

Default Messages (Low)	In below mentioned url, if add a specific payloadit will show default message. **Affected URL :** • http://13.126.196.134/?includelang=lang/en.php **Payload** • en.php' (GET Parameter)

Observation

By using the payload this error message is displayed. While showing the error we can also see the folders and code filename where the particular code is present.

Although this does not seem to be a big vulnerability but the names of files and folders can be of use for an attacker.



Business Impact - Moderate

As already seen in the previous slide this vulnerability does not have a direct impact to users or the server, though it can help the attacker in mapping the server architecture and plan further attacks on the server.

Recommendation

- Do not display the default error messages because it not tells about the server but also sometimes about the location.
- So, whenever there is an error, send it to the same page or throw some manually written error.

References:

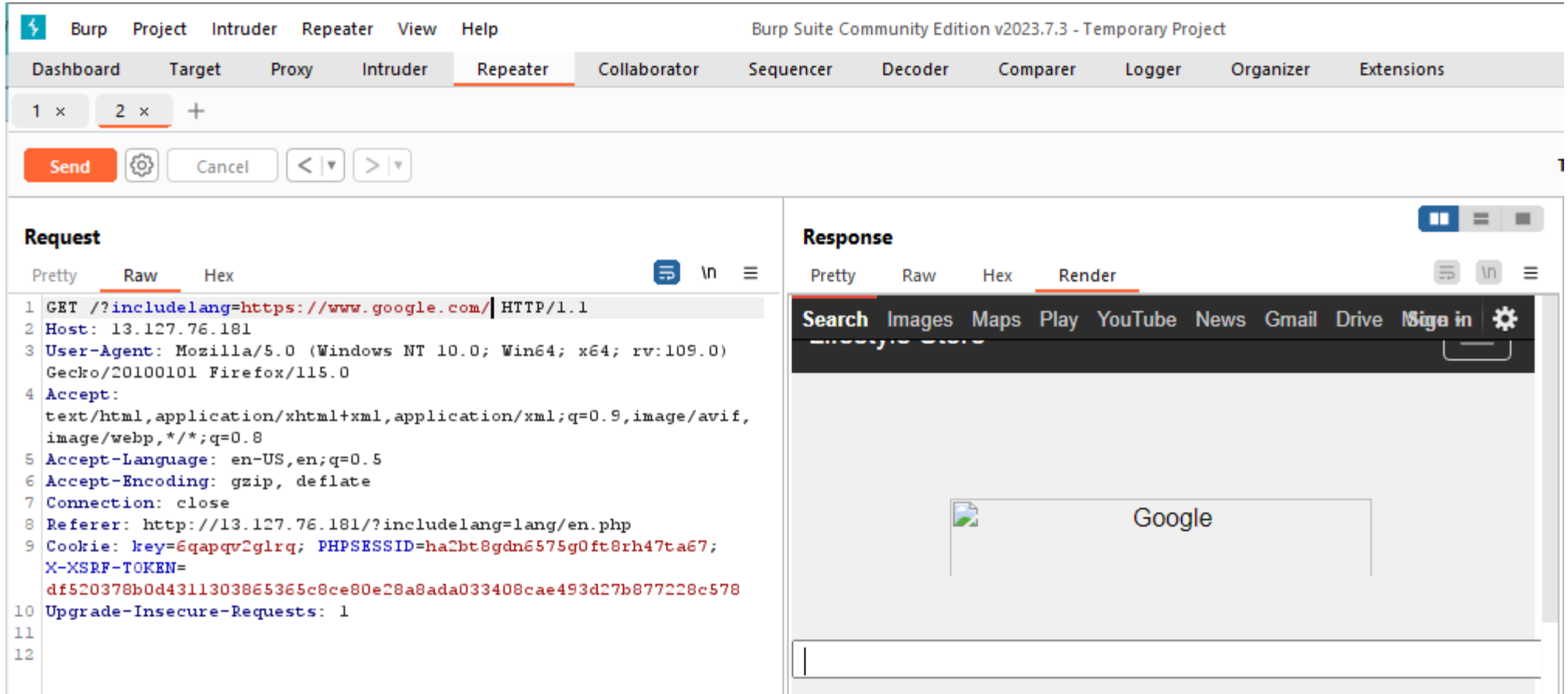
- https://www.owasp.org/index.php/Improper_Error_Handling

18. Open Redirection

Open Redirection (Low)	In below mentioned urls we can change the path or we are able to redirect the user to our intended location and not the website any URL. **Affected URL :** • http://13.127.76.181/?includelang=lang/en.php • http://13.127.76.181/?includelang=lang/fr.php **Payload:** • http://13.127.76.181/?inclludelang=https/www.google.com?lang/en.php

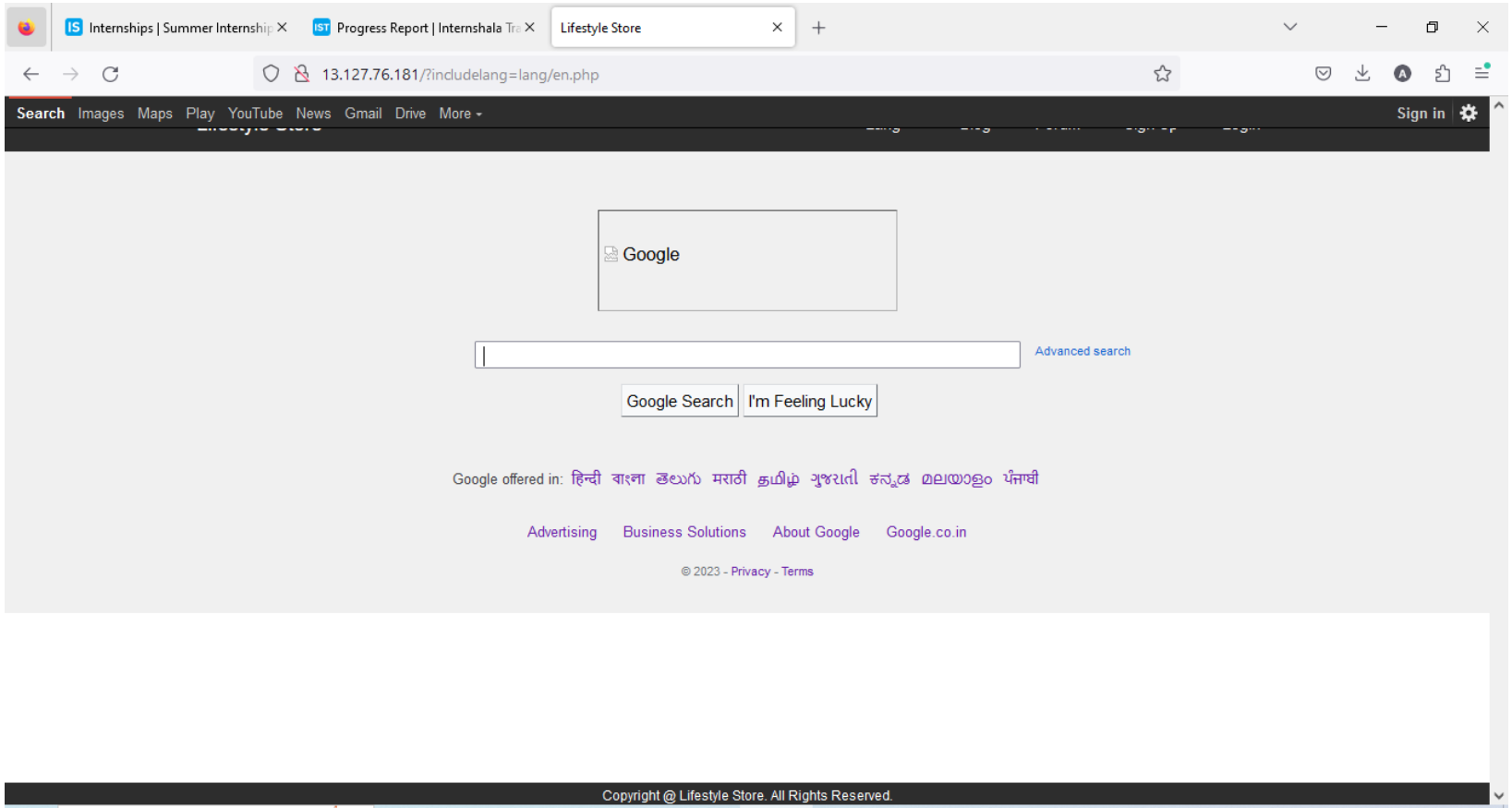
Observation

Here I intercepted the URL request and made changes into the URL such that once the user clicks on the button, instead of switching language, the page will be redirected to google.com



Proof Of Concept - POC

Here you can see that when the user pressed the English button under the lang section, the page will be redirected to google.com



Business Impact - Moderate

An http parameter may contain a URL value and could cause the web application to redirect the request to the specified URL. By modifying the URL value to a malicious site.

Thus this vulnerability could lend the website user to some unintended and malicious page, where there are potentially high charges of them getting duped in some kind of fraud.

This overall is going to affect the organization trust and reputation,

Recommendation

- Disallow Offsite Redirects.
- If you have to redirect the user based on URLs, instead of using untrusted input you should always use an ID which is internally resolved to the respective URL.
- If you want the user to be able to issue redirects you should use a redirection page that requires the user to click on the link instead of just redirecting them.
- You should also check that the URL begins with http:// or https:// and also invalidate all other URLs to prevent the use of malicious URIs such as javascript.

References:

- https://cwe.mitre.org/data/definitions/601.html
- https://www.hacksplaining.com/prevention/open-redirects

THANK YOU

For any further clarifications/patch assistance, please contact:
anasnirban00@gmail.com