

E-COMMERCE WEBSITE LIFESTYLE STORE

DETAILED DEVELOPER REPORT

SECURITY STATUS – EXTREMELY VULNERABLE

- Hackers can steal all the records of Lifestyle store(SQLi)
- Hacker can take control of complete server including View, Add, Edit, Delete files and folders.(shell upload and weak passwords)
- Hacker can change source code of application to host malware, phishing pages or even explicit content.(Shell upload)
- Hacker can see details of any customer.(IDOR)
- Hacker can easily access or bypass admin account authentication.(bruteforcing)
- Hacker can get access to seller details and login into the website using customer of the month usernames (PII).
- Hacker can change the password , confirm order and remove item of customer(CSRF)

VULNERABILITY STATISTIC



VULNERABILITIES:

S.NO	SEVERITY	VULNERABILITY	COUNT
1	CRITICAL	SQL injection	3
2	CRITICAL	Access to admin panel	1
3	CRITICAL	Arbitrary file upload	2
4	SEVERE	Reflected cross site scripting	1
5	SEVERE	Stored cross site scripting	1
6	SEVERE	Common password	1
7	SEVERE	Component with known vulnerability	3
8	MODERATE	Server misconfiguration	1
9	MODERATE	Unauthorized access to user details (IDOR)	4
10	MODERATE	Directory listings	5
11	LOW	Personal Information leakage	2
12	LOW	Client side and server side validation bypass	1
13	LOW	Default error display	1
14	LOW	Open redirection	2

1.SQL Injection

SQL Injection (Critical)

Below mentioned URL in the T-shirt/socks/shoes module is vulnerable to SQL injection attack

Affected URL :

- <http://13.232.247.247/products.php?cat=1>

Affected Parameters:

- cat (GET parameter)

Payload :

- cat=1'

Affected URL:

<http://13.232.247.247/products.php?q=socks>

Affected Parameters :

- q (GET parameter)

Payload:

- q=socks'

1.SQL Injection

SQL Injection
(Critical)

Here are other similar SQLi in the application
Affected URL :

- <http://13.232.247.247/products.php?cat=2>
- <http://13.232.247.247/products.php?cat=3>


OBSERVATION

Navigate to T-Shirt tab where you will see number of T-shirts. Notice the GET parameter CAT in the URL:


13.232.247.247/products.php?cat=1

Lifestyle Store

Blog Forum Sign Up Login ▾

Search 


[T Shirt](#) [Socks](#) [Shoes](#)



Basic T shirt


350

[VIEW PRODUCT](#)



Simple T Shirts

550



Plain Tee

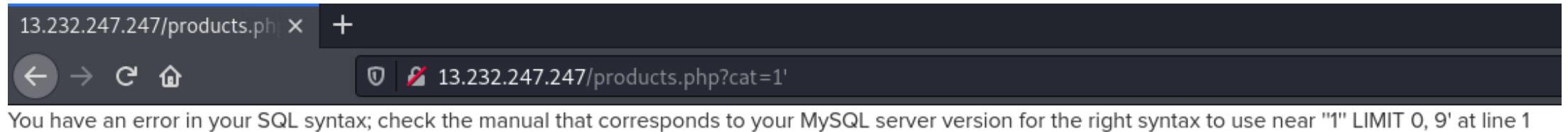
300

OBSERVATION

We apply single quote in cat parameter:

products.php?cat=1'

and we get complete MySQL error:



OBSERVATION

- We then put '--+' :
products.php?cat=1'--+
and the error is removed confirming SQL injection
- Now hacker can inject sql or use use sqlmap to get access to the database

Proof of Concept (PoC):- Attacker can dump arbitrary data

No of databases: 2

- information_schema
- hacking_training_project

No of tables : 10

1. Brands
2. cart_items
3. categories
4. customers
5. order_items
6. orders
7. product_reviews
8. products
9. sellers
10. user

user_name	password	email	unique_key
admin	\$2y\$10\$xkmdvrXSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	admin@lifestylestore.com	15468927955c66694cba1174.29688447
Donal234	\$2y\$10\$PM.7nBSP5FMaldXiM/S3s./p5xR6GTKvjry7ysJtxOkBqOJURAHsO	donald@lifestylestore.com	778522555c6669996f5a24.34991684
Pluto98	\$2y\$10\$xkmdvrXSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	Pluto@lifestylestore.com	19486318945c666a037b1432.99985767
chandan	\$2y\$10\$4cZBEIrgthXdvT1hwUlivuFELe03rR.GIcdp03NjrlS0VeiOKLVDa	chandan@lifestylestore.com	12404594545c666a3b49e0f8.08173871
Popeye786	\$2y\$10\$Fkv1RfwYTioW0w2CaZtAQuXVnhGAUjt/If/yTqkNPC5zTrsVm7EeC	popeye@lifestylestore.com	18430379145c666a53af8431.79566371
Radhika	\$2y\$10\$RYxNh0yV/G4g70tFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC.	radhika@lifestylestore.com	15611262655c666b312f73e0.70827297
Nandan	\$2y\$10\$G.cRNLMEiG79ZFxElhG.R.o95334U0xmZu4.9MqzR5614ucwnk59K	Nandan@lifestylestore.com	1587354115c666b65bb44a5.36505317
MurthyAdapa	\$2y\$10\$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1DPR.11UubDG	murthy@internshala.com	16357203785c68f640c699a2.83646347
john	\$2y\$10\$GhDB8h1X6XjPMY12GZ1vD07Y3en97u1/.oXTZLmYqB6F18FBgecvG	jhon@gmail.com	9946437385c6a435f76bef0.14675944
bob	\$2y\$10\$kiUikn3HPFbuyTtK75LLNurxzqC0LX3eMGy0/Uxl6JOoG37dCGKLq	bob@building.com	4305822125c6a43ec507df0.68309267
jack	\$2y\$10\$z/nyNlKRJ76m9ItMZ4N5l0eRxy6Gkqi9N/UBcJu5Ze07eM7N4pTHu	jack@ronald.com	15257114565c6a444692b707.17903432
bulla	\$2y\$10\$HT5oiRMetqaZ7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG	bulla@ranto.com	18292501185c6a4493a5ddb0.87138000
hunter	\$2y\$10\$pB3U9iFxbBgSbl2AkBpiEeIBdhiYfWy9y.xV23q12gGbMCyn7N3g2	konezo@web-experts.net	13824560345c80704e821145.26019698
asd	\$2y\$10\$At5pFZnRwpjCD/yNnJWDL.L3Cc4Cv0W8Q/WEHmWzBFqVikBQFpCF2	asd@asd.com	8057400125c862a7f5916c9.06111587
acdc	\$2y\$10\$J50B78.gpucuLTwpHwbcPedYcain.Yi.tsTLyQtK17FzdSpmIRRbi	cewi@next-mail.info	13104802695c86f43f0c3705.77019309

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. Previous slide has the screenshot of users table which shows user credentials being leaked that too in plain text without any hashing/encryption. 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.

RECOMENDATIONS

- 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)

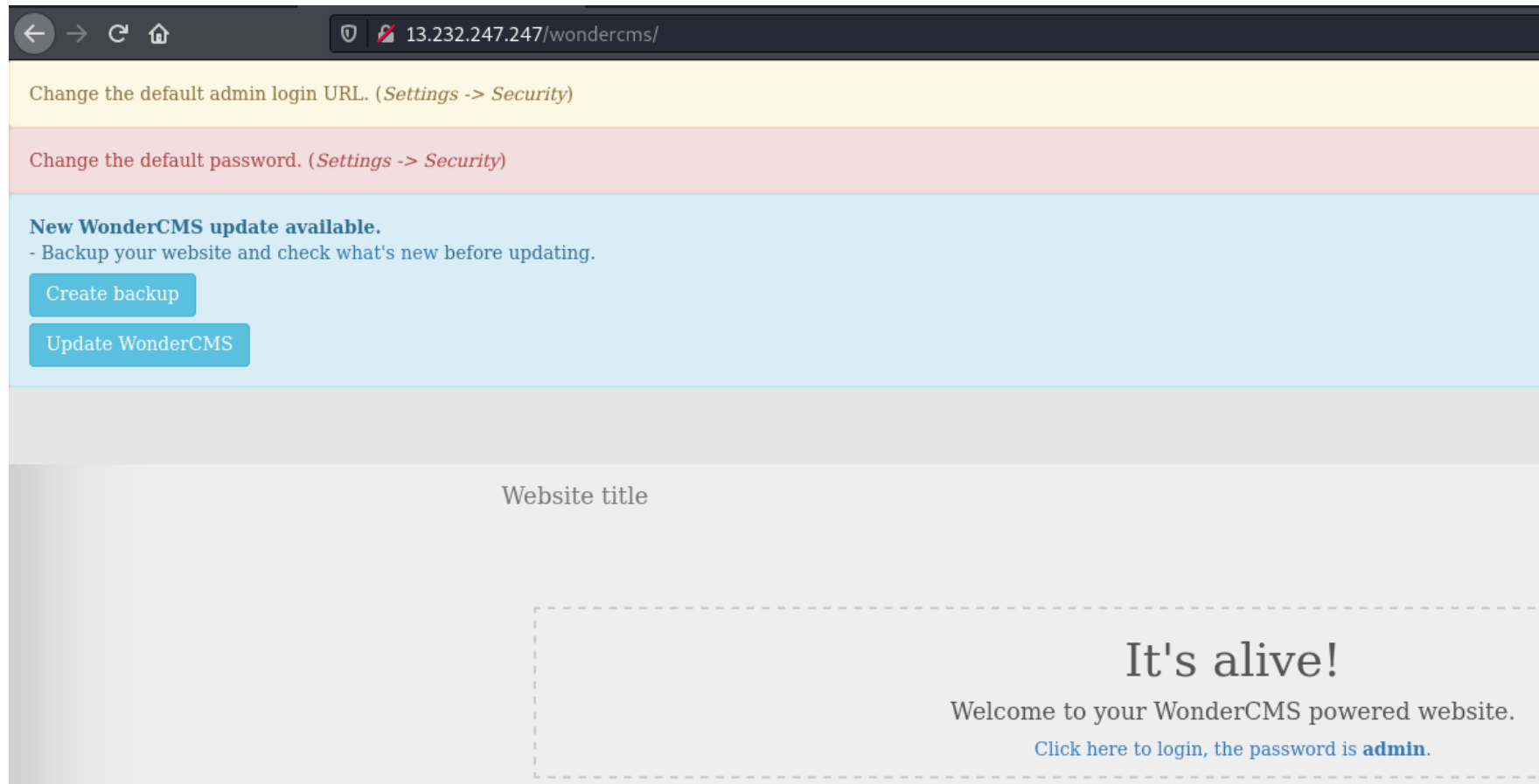
Below mentioned URL is vulnerable to Arbitrary File Upload and making other admin level changes.

Affected URL :

- <http://13.232.247.247/wondercms/loginURL>

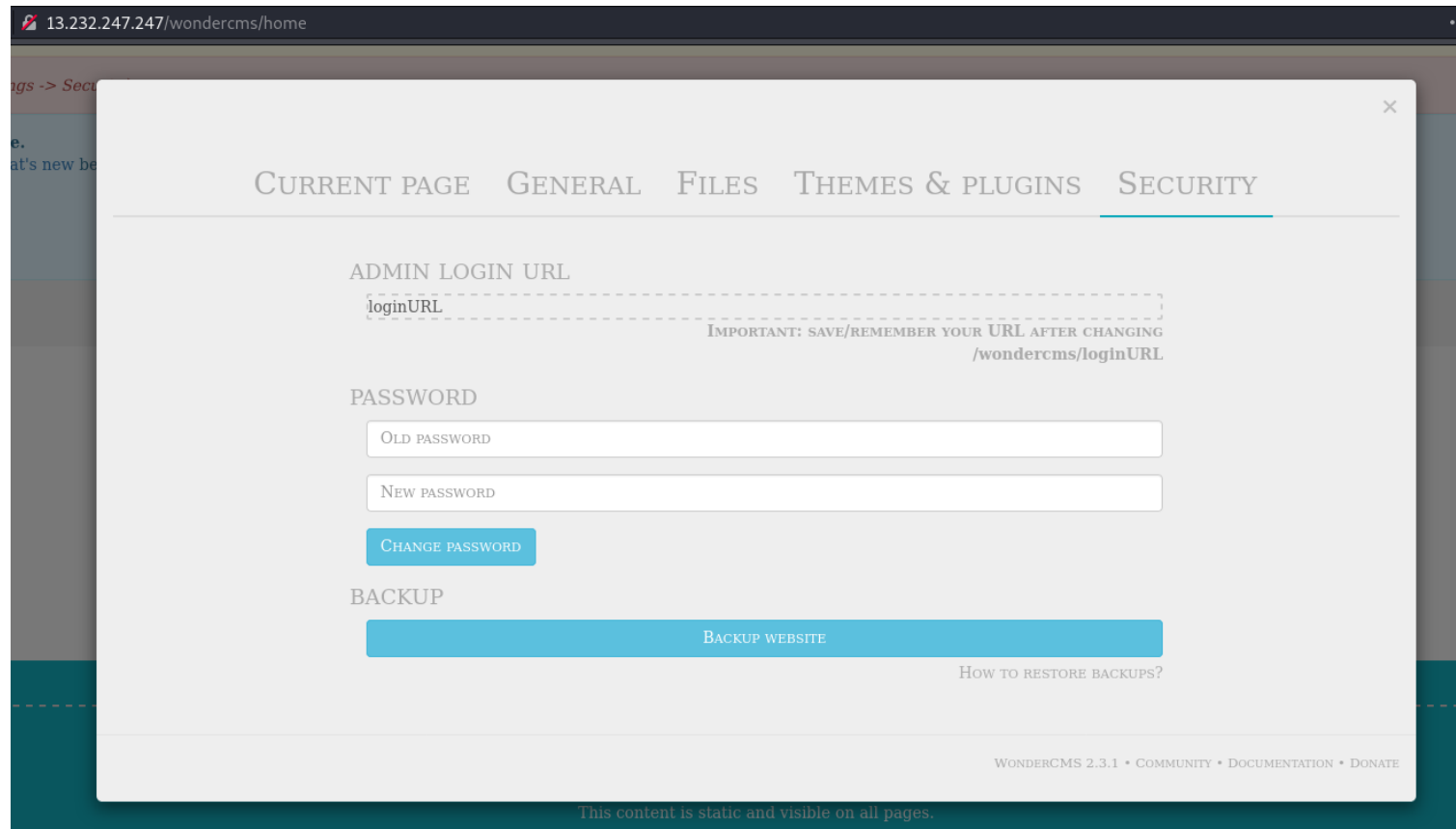
Observation

- When we navigate to <http://13.232.247.247/wondercms/loginURL/> url
- we get the password on the page and login as : admin in the url http://13.232.247.247/wondercms/loginURL



Proof of Concept (PoC)

- Hacker can change the admin password.
- Hacker can also add and delete pages.
- Hacker can upload any malicious file.



The screenshot shows a web browser window with the address bar displaying "13.232.247.247/wondercms/home". The main content area is a modal window titled "Security" with a close button (X) in the top right corner. The modal has a navigation bar with tabs: "CURRENT PAGE", "GENERAL", "FILES", "THEMES & PLUGINS", and "SECURITY". The "SECURITY" tab is selected and underlined. Below the tabs, the "ADMIN LOGIN URL" section contains a dashed box with the text "loginURL" and a note: "IMPORTANT: SAVE/REMEMBER YOUR URL AFTER CHANGING /wondercms/loginURL". The "PASSWORD" section has two input fields: "OLD PASSWORD" and "NEW PASSWORD", followed by a blue button labeled "CHANGE PASSWORD". The "BACKUP" section has a blue button labeled "BACKUP WEBSITE" and a link "HOW TO RESTORE BACKUPS?". At the bottom of the modal, there is a footer with the text "WONDERCMS 2.3.1 • COMMUNITY • DOCUMENTATION • DONATE". Below the modal, a footer bar contains the text "This content is static and visible on all pages."

Business impact - Extremely High

- Hacker 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

RECOMENDATIONS

- The default password should be changed and a strong password must be setup.
- 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/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 :

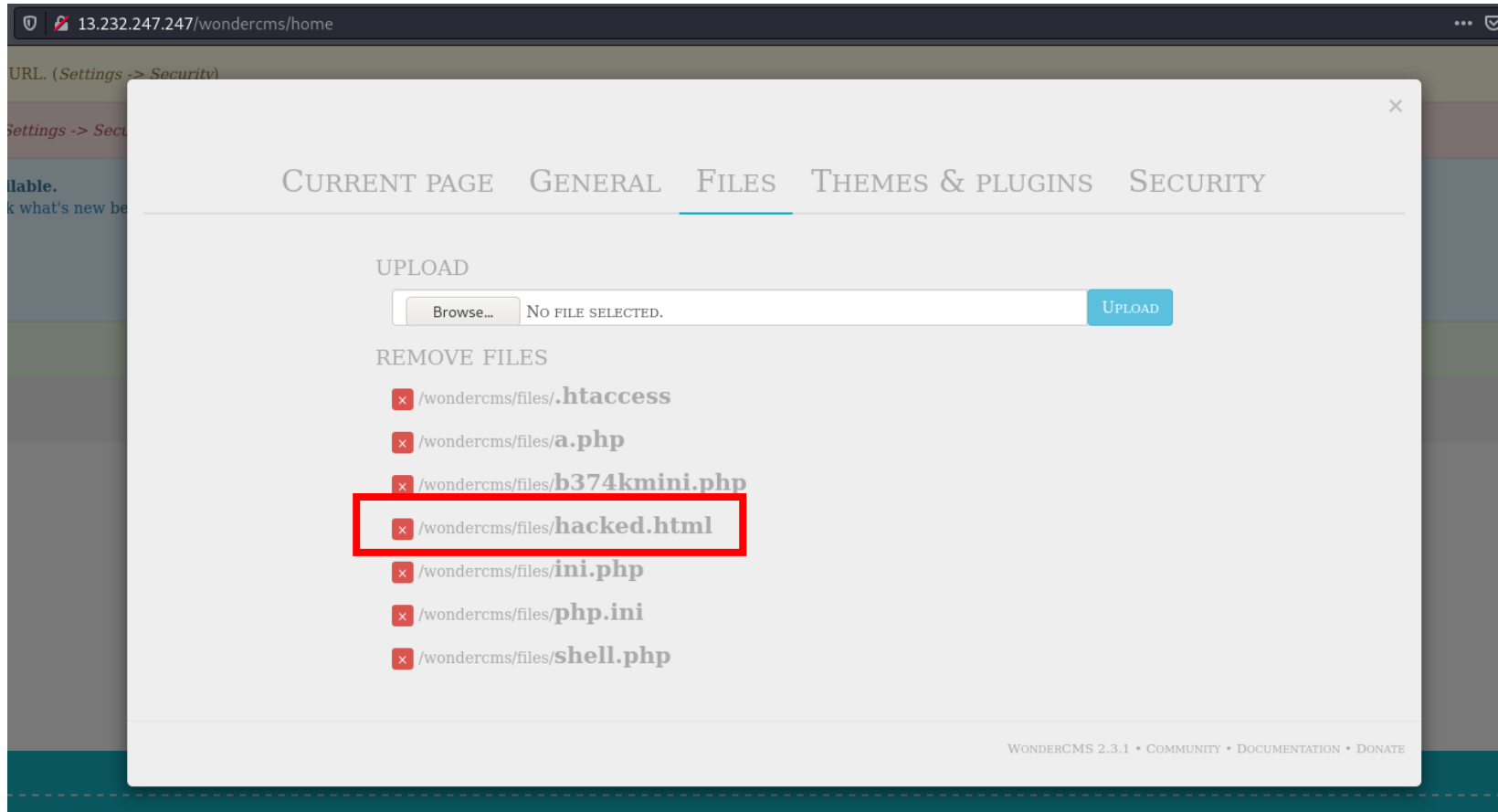
- <http://13.126.196.134/wondercms/>Affected Parameters :
- File Upload (POST parameter)

The attacker can upload files with extension other than .jpeg .

Affected URL :

- <http://13.126.196.134/profile/2/edit/> Affected Parameters :
- Upload Profile Photo (POST parameter)

Observation



Proof of Concept

- Weak password - admin.
- Arbitrary File Inclusion.

Business Impact – Extremely High

A malicious user can access the Dashboard which discloses many critical information of organization including:

- Important files
- Password
- And much more...

Business Impact – Extremely high

- Any backdoor file or shell can be uploaded to get access to the uploaded file on remote server and data can be exfiltrated. The presence of an actual malicious file can compromise the entire system leading to system takeover/ data stealing.

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>

Recommendation

Take the following precautions:

- Use a strong password 8 character or more in length with alphanumerics and symbols
- It should not contain personal/guessable information
- Do not reuse passwords
- Disable default accounts and users
- Change all passwords to strong unique passwords

References:

- [https://www.owasp.org/index.php/Testing_for_weak_password_change_or_reset_functionalities_\(OTG-AUTHN-009\)](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>

6. Reflected Cross Site Scripting (XSS)

Reflected Cross
Site Scripting
(Severe)

Below mentioned parameters are vulnerable to reflected XSS

Affected URL :

- <http://13.126.196.134/profile/16/edit/>

Affected Parameters :

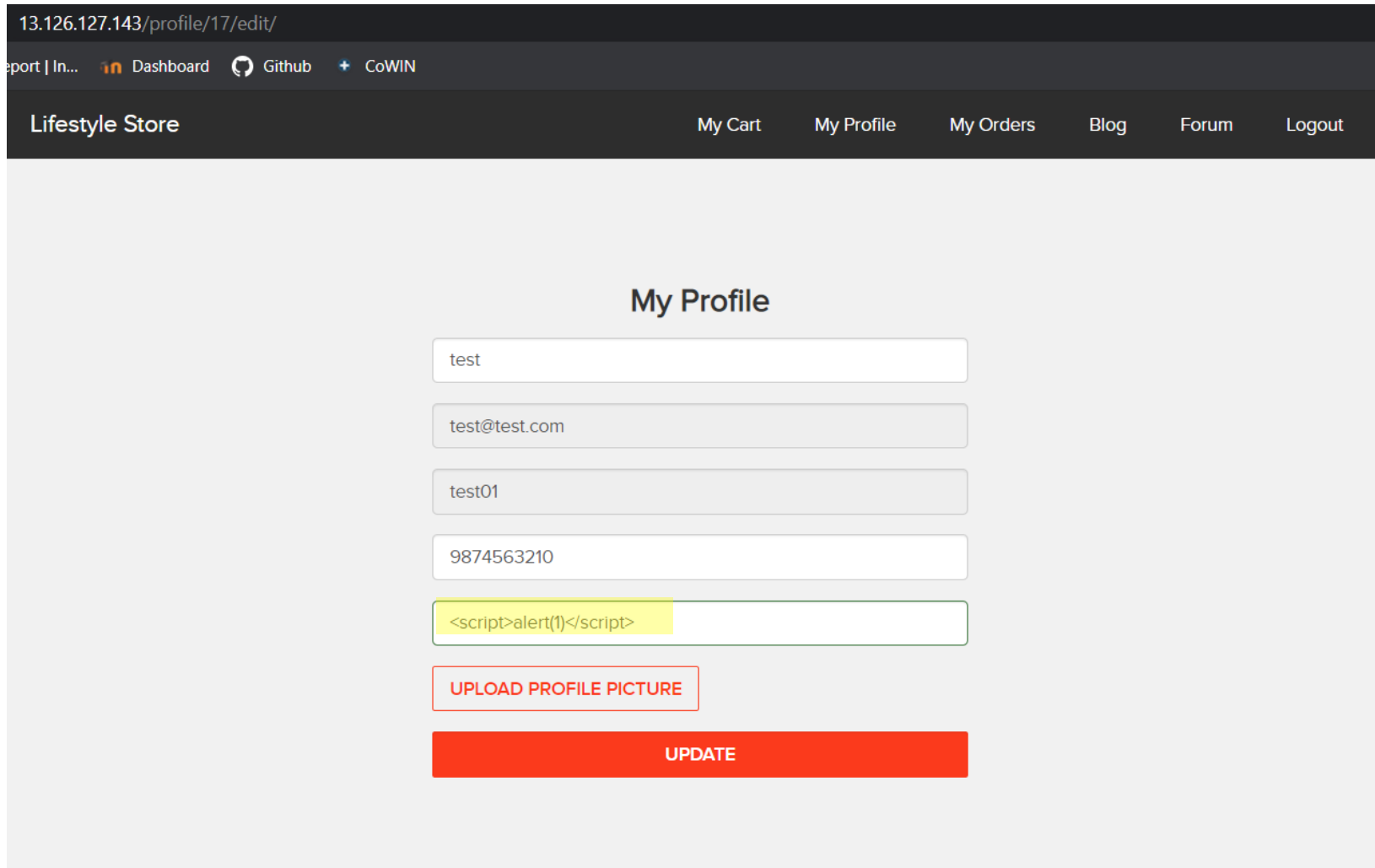
- address(POST parameters)

Payload:

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

Observation

Open edit profile through URL and write a script on address bar



The screenshot shows a web browser window with the address bar displaying `13.126.127.143/profile/17/edit/`. The browser's tab bar shows several tabs: "Report | In...", "Dashboard", "Github", and "CoWIN". The website's header is dark with the text "Lifestyle Store" on the left and navigation links "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout" on the right. The main content area is light gray and titled "My Profile". It contains five input fields: a text field with "test", an email field with "test@test.com", a text field with "test01", a text field with "9874563210", and a text field containing the JavaScript code `<script>alert(1)</script>`. Below the input fields is a red button labeled "UPLOAD PROFILE PICTURE" and a large red button labeled "UPDATE".

13.126.127.143/profile/17/edit/

Report | In... Dashboard Github CoWIN

Lifestyle Store My Cart My Profile My Orders Blog Forum Logout

My Profile

test

test@test.com

test01

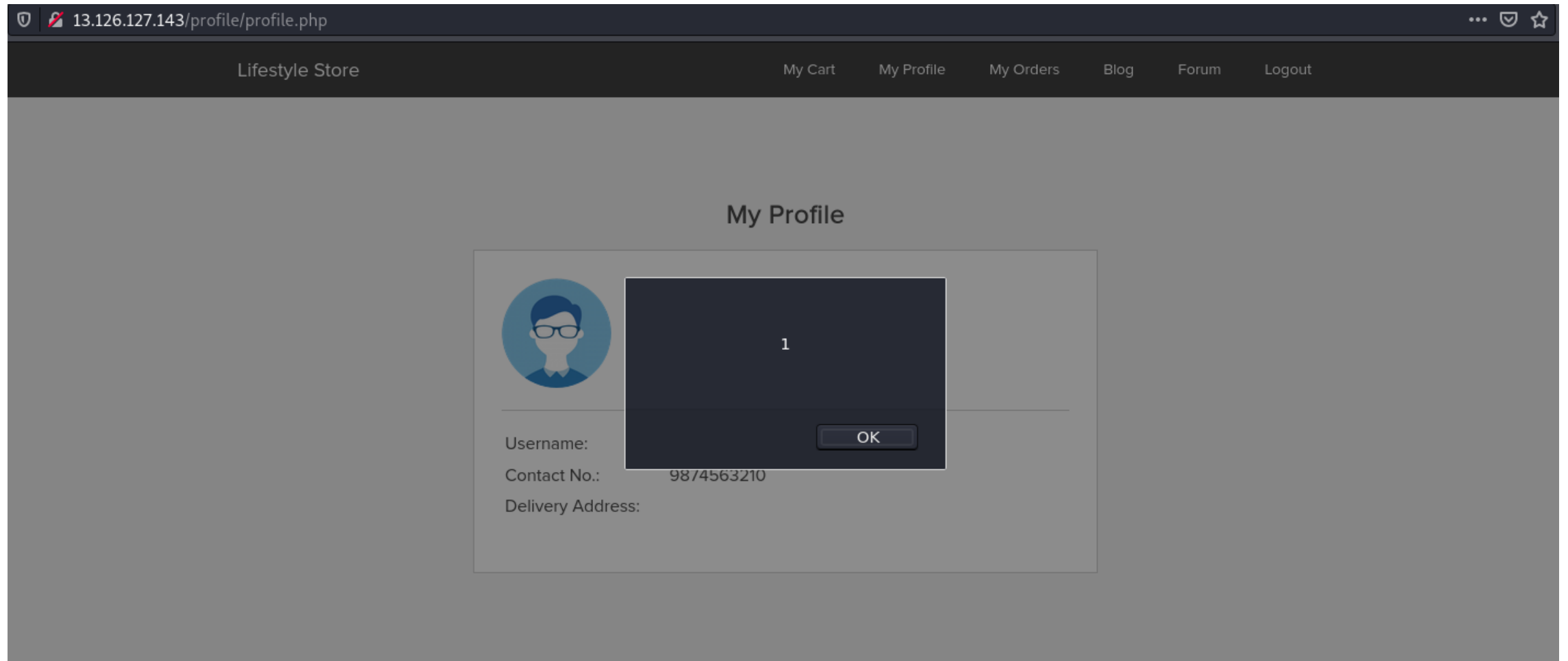
9874563210

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

UPLOAD PROFILE PICTURE

UPDATE

POC



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://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

7. Stored Cross Site Scripting (XSS)

Stored Cross
Site Scripting
(Severe)

Below mentioned parameters are vulnerable to reflected XSS

Affected URL :

- http://13.123.247.247/products/details.php?p_id=14

Affected Parameters :

- POST button under Customer Review (POST parameters)

Payloads:

- `<script>alert(Hacked)</script>`
- `<h1>Hey</h1>`

Observation

Now try entering the payload in review box

[All Products Socks](#)

Reebok Men Socks

Men Ankle Length Socks

[Seller Info](#)

[Brand Website](#)

INR 1111/-

Added

Customer Reviews



test

Hey



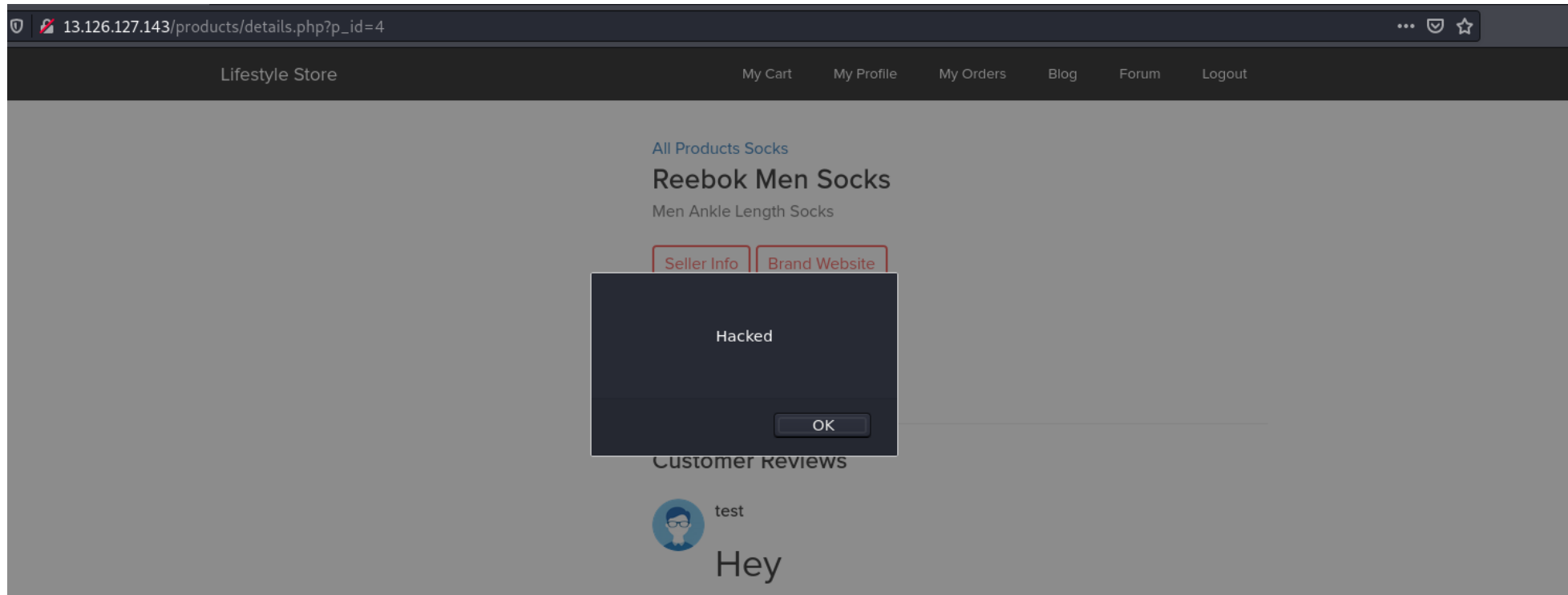
test

`<script>alert(Hacked)</script>`

POST

Observation

Hit post button , you can see stored XSS or permanent XSS



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:

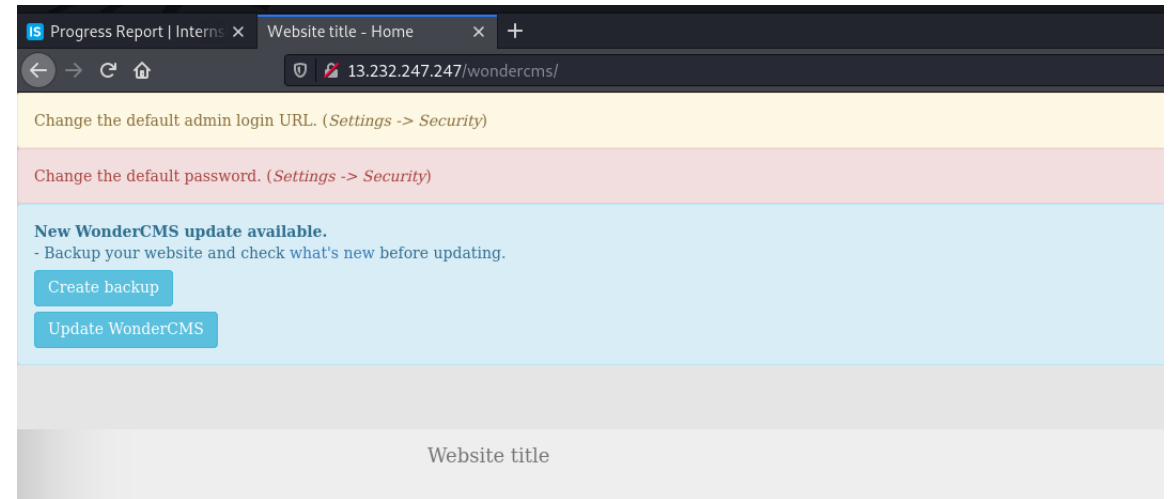
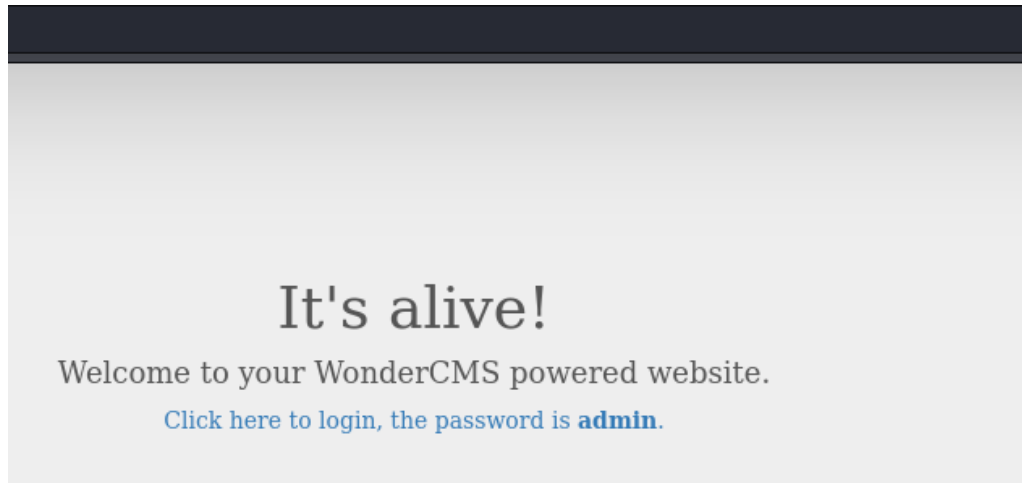
- [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- https://en.wikipedia.org/wiki/Cross-site_scripting
- https://www.w3schools.com/html/html_entities.asp

8. COMMON PASSWORD

Common password (Severe)	<p>Below mentioned url has weak and very common password</p> <p>Affected URL :</p> <ul style="list-style-type: none">• http://13.123.247.247/wondercms/

Observation

- Password is right in front of you



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:

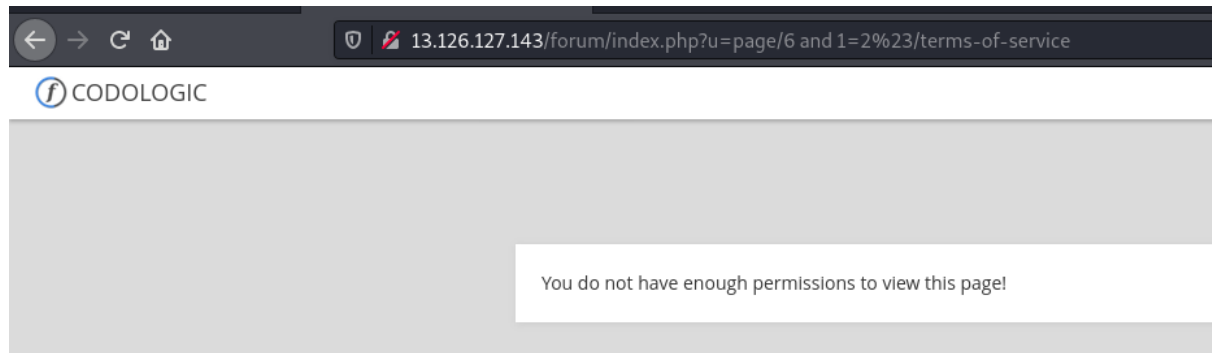
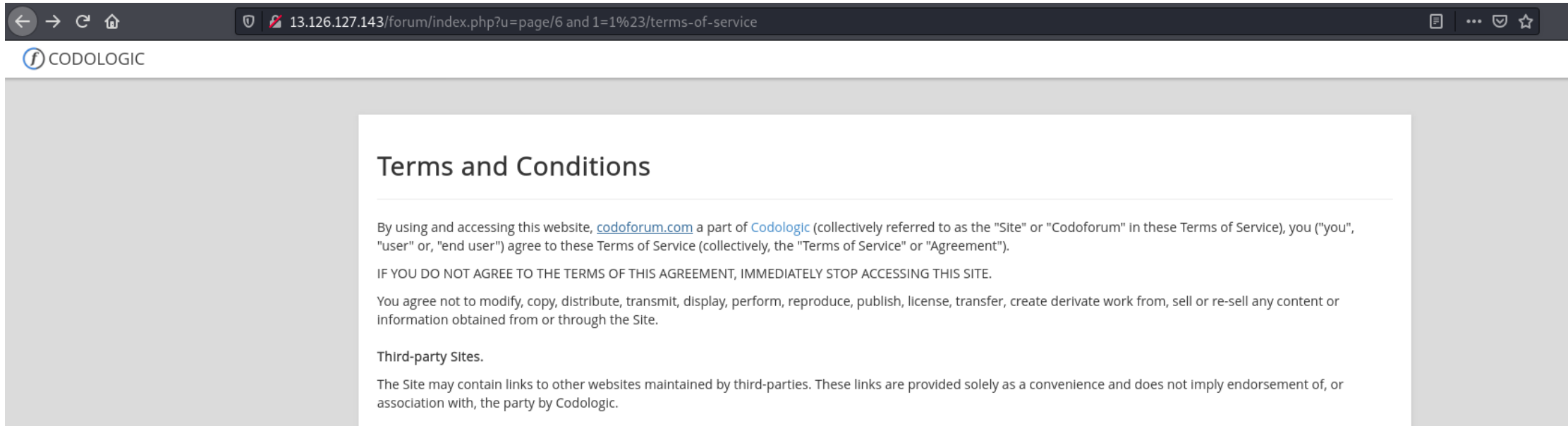
<https://www.acunetix.com/blog/articles/weak-password-vulnerability-common-think/>
[https://www.owasp.org/index.php/Testing_for_Weak_password_policy_\(OTG-AUTHN-007\)](https://www.owasp.org/index.php/Testing_for_Weak_password_policy_(OTG-AUTHN-007))

9. Component with known vulnerability

Component with
known
vulnerability
(Severe)

- Server used is nginx/1.14.0 appears to be outdated (current is at least 1.17.3) i.e it is known to have exploitable vulnerabilities.
- WonderCMS
- Codoforum (Powered by codologic)

Observation



POC

Codologic Vulnerability, It has multiple sql injection vulnerability, Check the link of exploit-db in reference.

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 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.

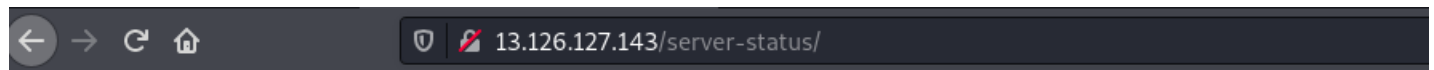
References:

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

10. Server misconfiguration

Component with known vulnerability (Severe)	<p>Below mentioned url will show you the server related info</p> <p>URL</p> <ul style="list-style-type: none">• http://13.126.127.143/server-status/

Observation and 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

-----w_-----
.....
.....

Scoreboard Key:

"_" Waiting for Connection, "s" Starting up, "R" Reading Request,
"w" Sending Reply, "k" Keepalive (read), "D" DNS Lookup,
"c" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	1709	0/1/1	_	0.92	17771	89	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET / HTTP/1.1
0-0	1709	0/1/1	_	9.64	34	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
0-0	1709	0/1/1	_	9.58	170	0	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /favicon.ico HTTP/1.1
0-0	1709	0/1/1	_	9.65	26	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
0-0	1709	0/1/1	_	9.66	16	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1

Recommendation

- Keep the software up to date
- Disable all the default accounts and change passwords regularly
- 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

References

<https://www.ifourtechnolab.com/blog/owasp-vulnerability-security-misconfiguration>

11. Unauthorized Access to User Details(IDOR)

Unauthorized
access to user
details
(Moderate)

Below mentioned url will have vulnerabilty through which anyone can see the details of another user

URL

http://13.126.127.143/orders/generate_receipt/ordered/10

Affected parameter
Ordered/**10**

Payload

http://13.126.127.143/orders/generate_receipt/ordered/11

11. Unauthorized Access to User Details(IDOR)

Unauthorized
access to user
details
(Moderate)

Below mentioned url will 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://13.127.159.1/orders/orders.php?customer=13/>
- <http://13.127.159.1/profile/16/edit/>
- <http://13.127.159.1/forum/index.php?u=/user/profile/4>

Observation

- When we change the payload we can see the receipts of other users or customers

13.126.127.143/orders/generate_receipt/ordered/10

Lifestyle StoreMy CartMy ProfileMy OrdersBlogForumLogout

Receipt

Order Id: 2DD930939259	
PRODUCTS:	
Adidas Socks - Pack	INR 450
Total	INR 450
SHIPPING DETAILS:	PAYMENT MODE
Name - asd	Cash on delivery
Email - asd@asd.com	
Phone - 9876543210	
Address - asdasd	
Order placed on : 2019-03-11 15:15:24	Status: DELIVERED

POC

- Here you can clearly see the receipt of another user

The screenshot shows a web browser window with the address bar displaying `13.126.127.143/orders/generate_receipt/orderId/11`. The page title is "Lifestyle Store". The navigation menu includes "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout". The main content area is titled "Receipt" and contains a white box with the following details:

Order Id: 7A9F72CC37AF	
PRODUCTS:	
Reebok Men Socks	INR 1111
Total	INR 1111
SHIPPING DETAILS:	PAYMENT MODE
Name - test	Cash on delivery
Email - test@test.com	
Phone - 9874563210	
Address - alert(1)	
Order placed on : 2021-07-09 13:47:56	Status: DELIVERED

Business Impact – Extremely 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

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/blackmarket. More over, as there is no ratelimiting 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

Take the following precautions:

- 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

12 Directory Listings

Directory
listings
(Moderate)

Below mentioned urls disclose server information.
Affected URL :

- <http://13.126.127.143/phpinfo.php>
- <http://13.126.127.143/robots.txt>
- <http://13.126.127.143/composer.lock>
- <http://13.126.127.143/composer.json>
- <http://13.126.127.143/urerlist.tx>

Observation

13.126.127.143/phpinfo.php



13.126.127.143/robots.txt

User-Agent: *
Disallow: /static/images/
Disallow: /ovidientiaCMS

PHP Version 5.6.39-1+ubuntu18.04.1+deb.sury.org+1



System	Linux ip-172-26-7-207 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
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar

POC

- In above observation you can see that a hacker can go through these directory easily and gather as much as information he/she want.
- Infact it also shows some accounts of seller



```
Radhika:Radhika123:6  
Nandan:Nandan123:7  
chandan:chandan123:4
```

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

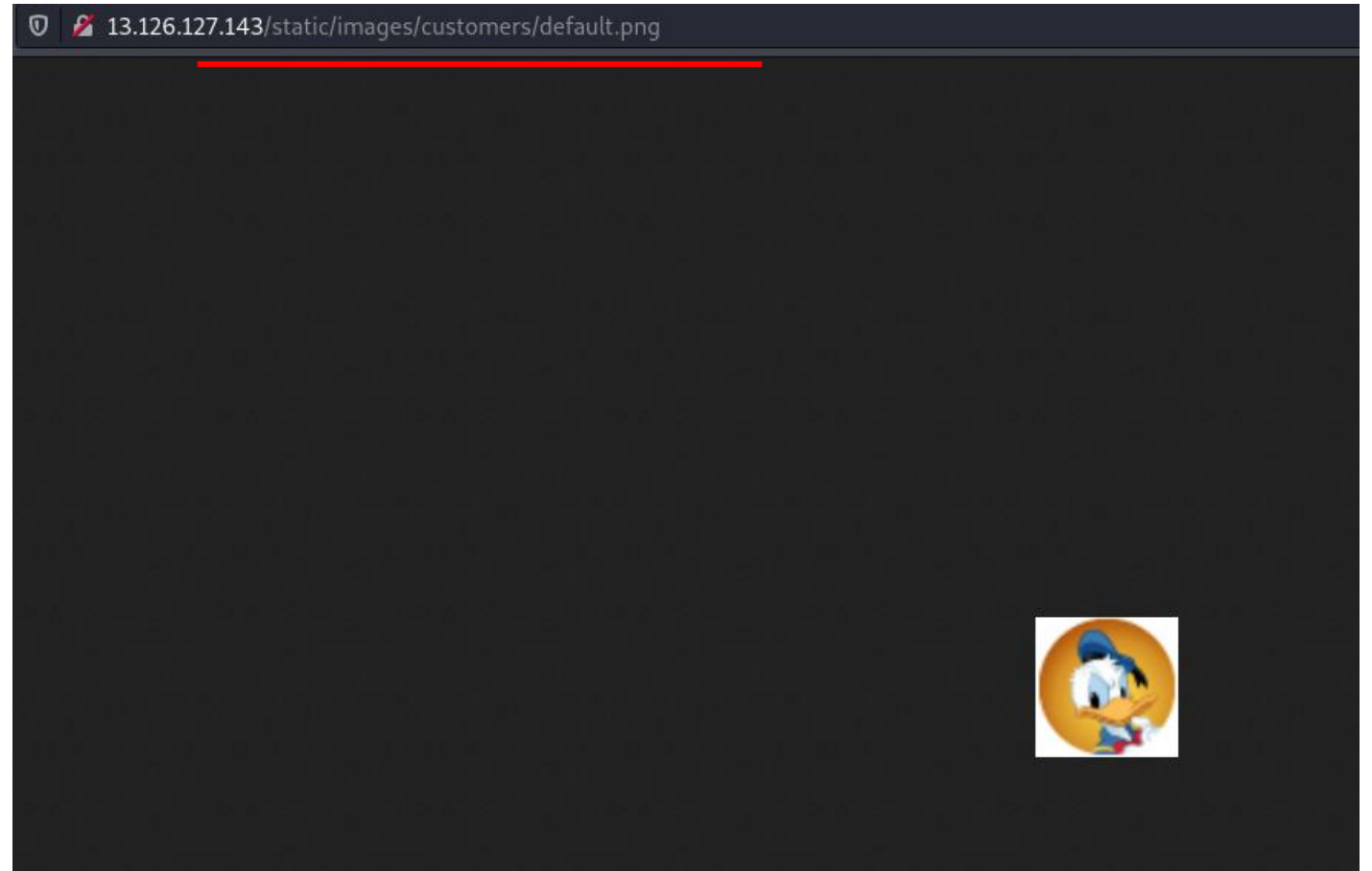
- <https://www.netsparker.com/blog/web-security/information-disclosure-issues-attacks/>
- <https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/information-disclosure-phpinfo/>

13. Personal Information Leakage

Personal Information Leakage (Low)	<p>Below mentioned urls disclose personal information</p> <p>Affected URL :</p> <ul style="list-style-type: none">• http://13.126.127.143/static/images/customers/default.pn• http://13.126.127.143/products/details.php?p_id=2

Observation

- Navigate to mentioned URL
- And you can see the whole path where everyone's photo is stored



POC

- Here if you see the url , you will know that we just changed it little bit and we hit jackpot where we can see photos uploaded by customer and may more...

13.126.127.143/static/images/uploads/			
Index of /static/images/uploads/			
../			
customers/	07-Jan-2019 08:49	-	
products/	07-Jan-2019 08:49	-	
card.png	05-Jan-2019 06:00	91456	

13.126.127.143/static/images/uploads/customers/			
Index of /static/images/uploads/customers/			
../			
1550224525.png	15-Feb-2019 09:55	10194	
1550228019.jpg	15-Feb-2019 10:53	9796	
1550382697.jpg	17-Feb-2019 05:51	14616	
1550382890.jpg	17-Feb-2019 05:54	180769	
1552082680.jpg	08-Mar-2019 22:04	178491	
1552082706.jpg	08-Mar-2019 22:05	178491	
1552083012.jpg	08-Mar-2019 22:10	32935	
1552083459.jpg	08-Mar-2019 22:17	58	
default.png	07-Jan-2019 08:49	43218	

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

Recommendations

- 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>

14. Client side and server side validation bypass

Client side and
server side
validation by
pass (Low)

In below mentioned urls , we can easily bypass client side and server side validation

Affected URL :

- <http://13.126.127.143/profile/17/edit/>

Affected parameter:

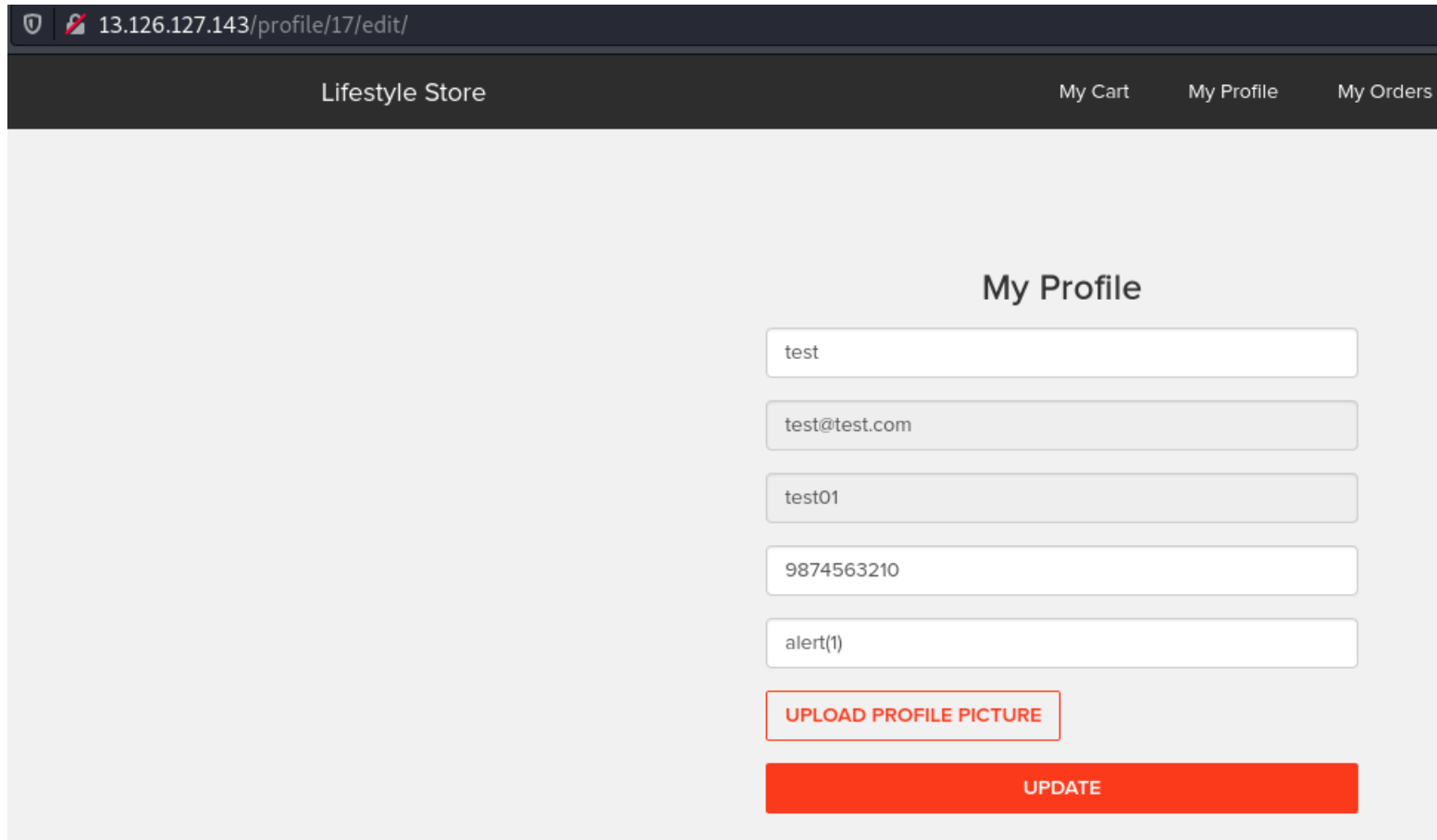
- Contact Number (POST Parameter)

Payload used:

- 0000000000

Observation

Here we intercepted the request and made changes in the contact number field



The screenshot shows a web browser window with the address bar displaying `13.126.127.143/profile/17/edit/`. The page header includes the text "Lifestyle Store" and navigation links for "My Cart", "My Profile", and "My Orders". The main content area is titled "My Profile" and contains several input fields. The first field contains "test", the second "test@test.com", the third "test01", the fourth "9874563210", and the fifth "alert(1)". Below these fields is a red button labeled "UPDATE PROFILE PICTURE" and a large red button labeled "UPDATE".

13.126.127.143/profile/17/edit/

Lifestyle Store My Cart My Profile My Orders

My Profile

test

test@test.com

test01

9874563210

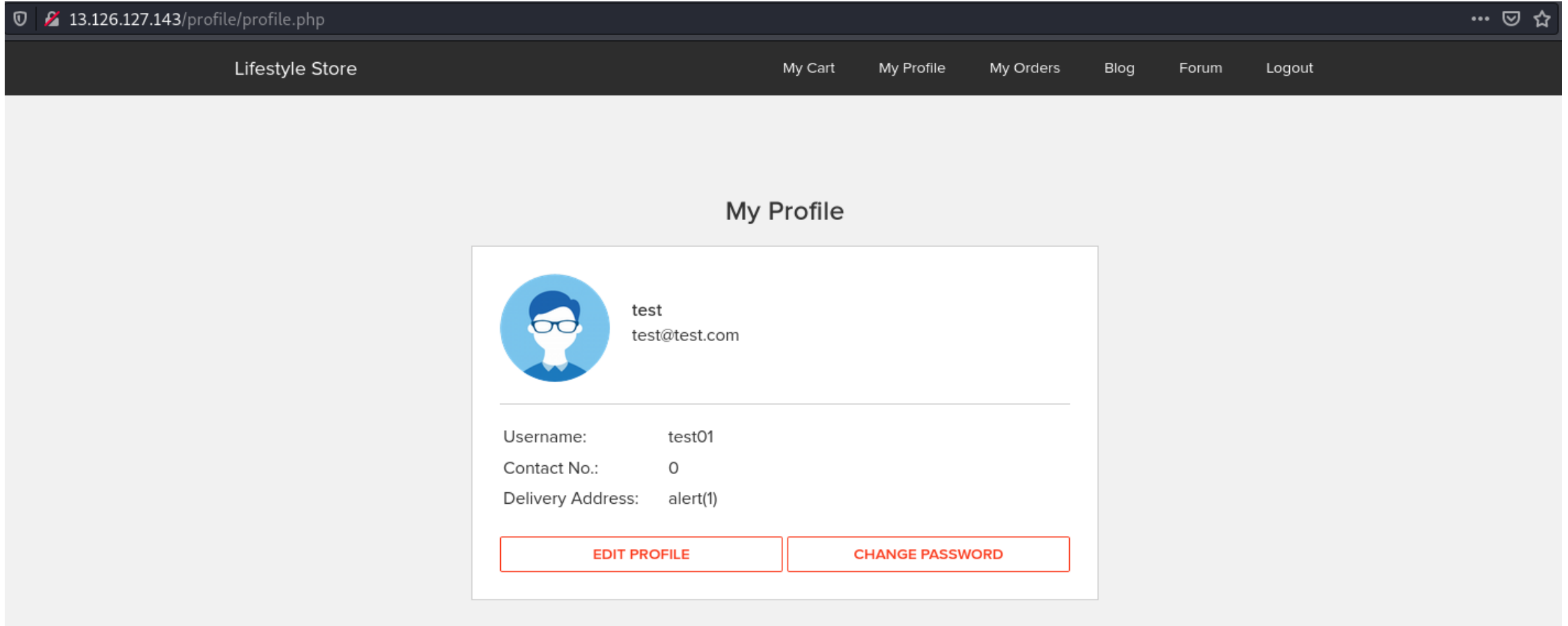
alert(1)

UPLOAD PROFILE PICTURE

UPDATE

POC

- Mobile number is saved as zero



Business Impact – Moderate

The data provided by the user ,if incorrect, is not a very big issue but still must be checked for proper validity information.

Recommendations

- Implement all critical checks on server side code only.
- Client-side checks must be treated as decoratives only.
- All business logic must be implemented and checked on the server code.

REFERENCES:-

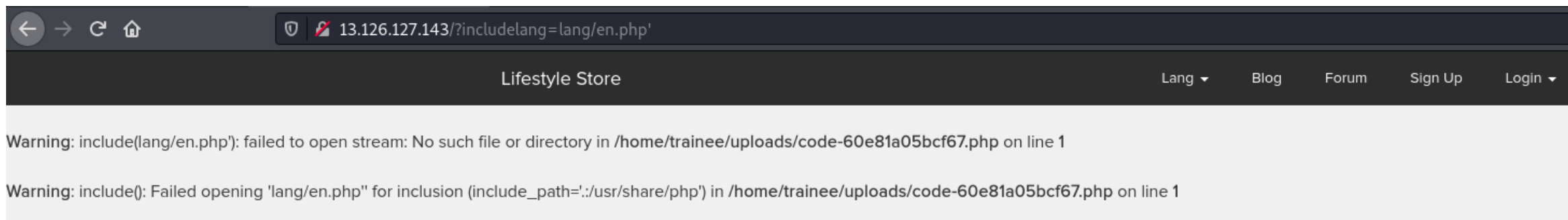
- <http://projects.webappsec.org/w/page/13246933/Improper%20Input%20Handling>
- https://www.owasp.org/index.php/Unvalidated_Input

15. Default Messages

Default messages (Low)	<p>In below mentioned urls ,if add a specific payload it will show default messages</p> <p>Affected URL :</p> <ul style="list-style-type: none">•http://13.126.127.143/?includelang=lang/en.php <p>Payload</p> <ul style="list-style-type: none">•en.php' (GET Parameter)

Observation & POC

Here we added payload as shown above and we got an error



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 server architecture and plan further attacks on the server.

Recommendations

- 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

16. Open Redirection

Open
Redirection
(Low)

In below mentioned urls we can change the path of redirection Affected URL :

- <http://13.126.127.143/?inclludelang=lang/en.php>
- <http://13.126.196.134/?inclludelang=lang/fr.php>

Payload:-

- <http://13.126.196.134/?inclludelang=https/www.google.com?lang/en.php>

Observation

Here we made changes to the url according to the payload

 Request to http://13.126.127.143:80

Forward

Drop

Intercept is on

Action

Open Browser

Comment this item



Pretty

Raw

\n

Actions

1

2

3

4

5

6

7

8

9

10

11

12

GET

/?includelang=http://www.google.com?lang/en.php

HTTP/1.1

Host:

13.126.127.143

User-Agent:

Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0

Accept:

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

Accept-Language:

en-US,en;q=0.5

Accept-Encoding:

gzip, deflate

Connection:

close

Referer:

http://13.126.127.143/?includelang=lang/en.php

Cookie:

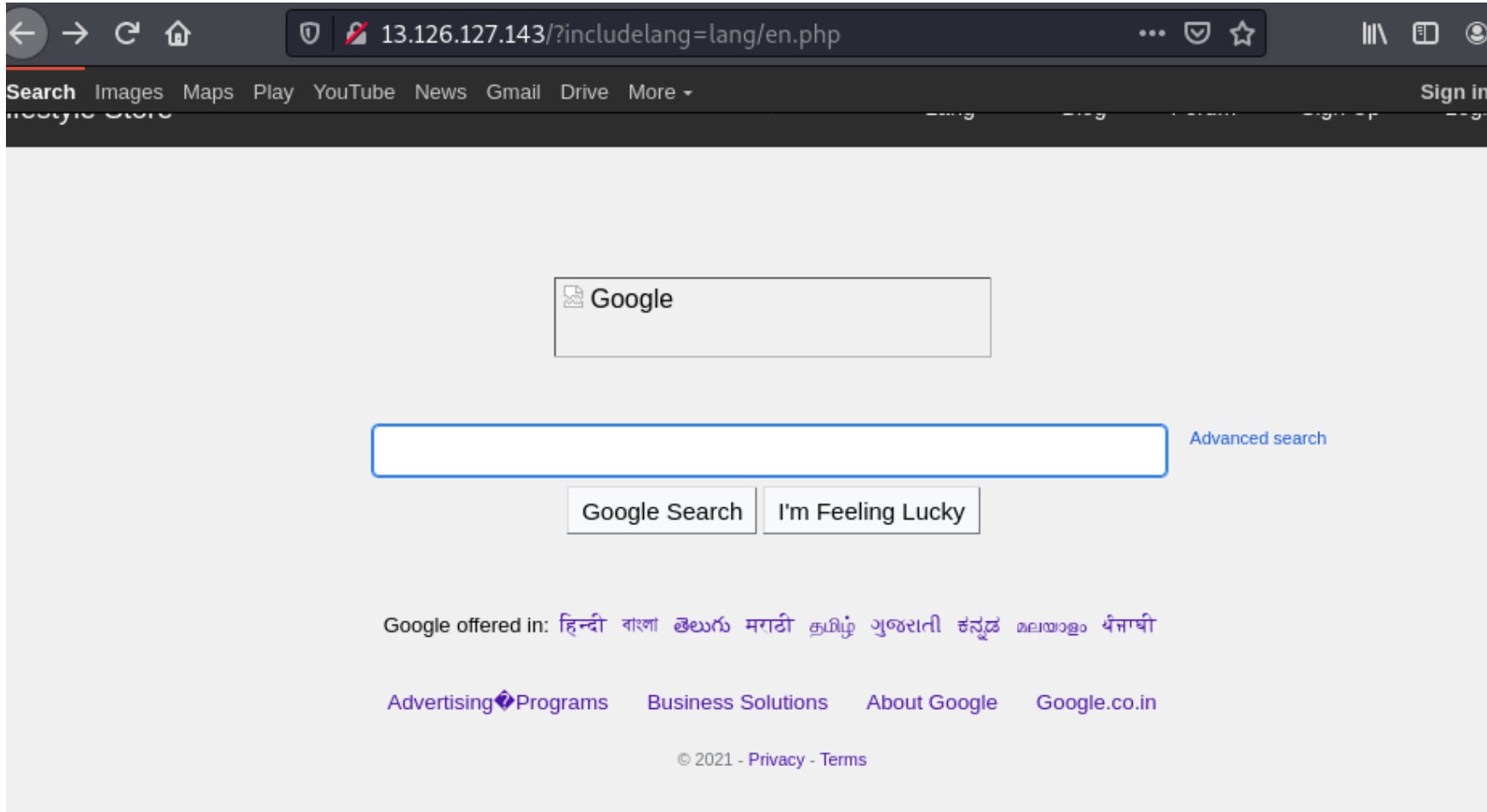
X-XSRF-TOKEN=957327742096ddaccb74eba6f36cc4f1eb72f94fedc3e6c593c13d321cde9d25; key=85BF48C5-9CB8-6BCE-683A-CBD41A0621ED; PHPSESSID=gilolllpvllqsr70dm6j7uadp2

Upgrade-Insecure-Requests:

1

POC

- We are redirected to google



Business Impact – low

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

Recommendations

- 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:
9876543210