VAPT Report For e-Commerce Platform.

“Lifestyle Store”
A shopping web application

Report By: Anup Adhikari

Introduction:

This is the vulnerability assessment and penetration testing(VAPT) report for “Lifestyle Store”.

I have reported all the vulnerabilities found on this store with their screenshots and links.

I have also provided business risk and expert recommendations to tackle those vulnerabilities.

This presentation contains a total of 59 slides, which are read only by default, you can opt-in for editing.

Security Status – Extremely Vulnerable

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

Vulnerability Statistics

Critical	Severe	Moderate
17	15	2

Low
2

Vulnerabilities:

No	Severity	Vulnerability	Count
1	Critical	SQL Injection	8
2	Critical	Access to sales dashboard	1
3	Critical	Access to admin panel	1
4	Critical	Account takeover via OTP Bypass	2
5	Critical	Unauthorized Access To Customer Details	5
6	Severe	Reflected cross site scripting	15
7	Moderate	Directory Listing of Configuration FIles	2
8	Low	Information disclosure due to Apache Default Pages	2

1. SQL Injection

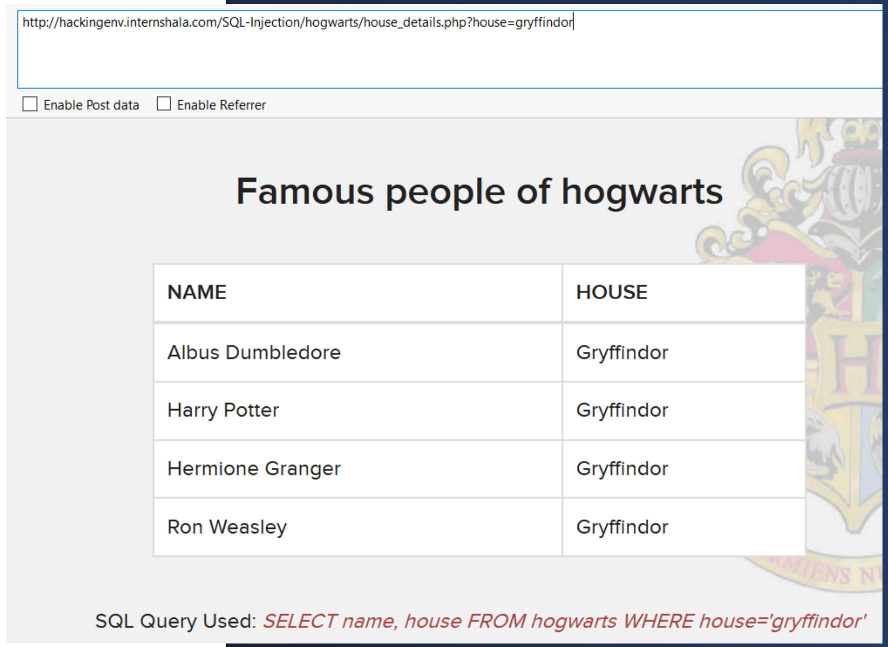
SQL Injection (Critical)	Below mentioned URL in the **Hogwarts House Details module** is vulnerable to SQL injection attack **Affected URL :** • http://url.com/hogwarts/house_details.php?house=HERE **Affected Parameters :** • house (GET parameter) **Payload:** • house=gryffindor’

1. SQL Injection

SQL Injection (Critical)	Here are other similar SQLi in the application **Affected URL :** • http://url.com/sql3.php (ID GET parameter) • http://url.com/sql4.php (jkl POST parameter) • http://url.com/sql5.php (pqr 5 GET parameter) • http://url.com/sql6.php (abcd cookie paramter) • http://url.com/sql7.php (User-agent Header) • http://url.com/sql8.php (xyz POST parameter)

Observation

- Navigate to Houses page where you will see list of houses. Click anyone like Gryffindor. You will see famous people of that house in a table. Notice the GET parameter **house** in the URL:



http://hackingenv.internshala.com/SQL-Injection/hogwarts/house_details.php?house=gryffindor

☐ Enable Post data ☐ Enable Referrer

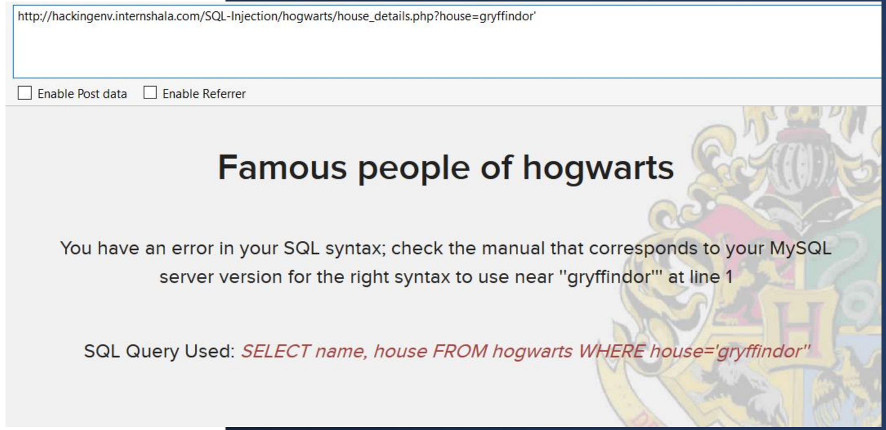
Famous people of hogwarts

NAME	HOUSE
Albus Dumbledore	Gryffindor
Harry Potter	Gryffindor
Hermione Granger	Gryffindor
Ron Weasley	Gryffindor

SQL Query Used: *SELECT name, house FROM hogwarts WHERE house='gryffindor'*

Observation

- We apply single quote in house parameter: **house_details.php?house=Gryffindor’ and we get complete MySQL error:**



```
http://hackingenv.internshala.com/SQL-Injection/hogwarts/house_details.php?house=gryffindor'
```

☐ Enable Post data ☐ Enable Referrer

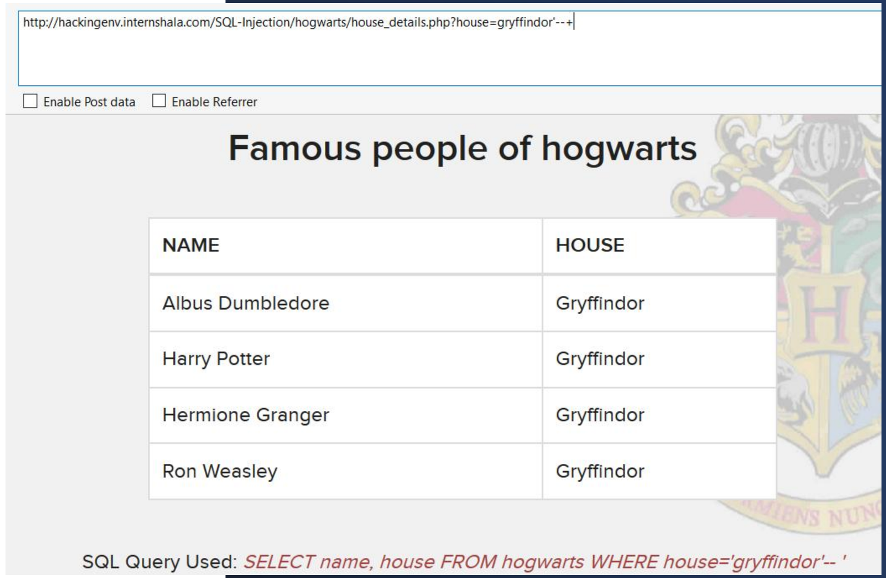
Famous people of hogwarts

```
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''gryffindor''' at line 1

SQL Query Used: SELECT name, house FROM hogwarts WHERE house='gryffindor''
```

Observation

- We then put --+ : **house_details.php?house=Gryffindor'--+ and we error is removed confirming SQL injection:**



```
http://hackingenv.internshala.com/SQL-Injection/hogwarts/house_details.php?house=gryffindor'--+
```

☐ Enable Post data ☐ Enable Referrer

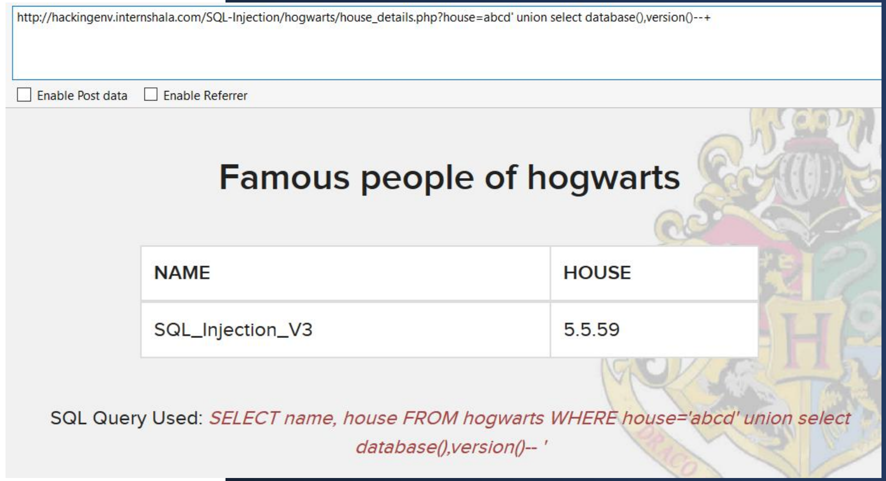
Famous people of hogwarts

NAME	HOUSE
Albus Dumbledore	Gryffindor
Harry Potter	Gryffindor
Hermione Granger	Gryffindor
Ron Weasley	Gryffindor

SQL Query Used: *SELECT name, house FROM hogwarts WHERE house='gryffindor'-- '*

Proof of Concept (PoC)

- Attacker can execute SQL commands as shown below. Here we have used the payload below to extract the database name and MySQL version information:
  house=abcd’ union select database(),version()--+



```
http://hackingenv.internshala.com/SQL-Injection/hogwarts/house_details.php?house=abcd' union select database(),version()--+
```

Famous people of hogwarts

NAME	HOUSE
SQL_Injection_V3	5.5.59

SQL Query Used: *SELECT name, house FROM hogwarts WHERE house='abcd' union select database(),version()-- '*

PoC – Attacker can dump arbitrary data

No of databases: 3

Information_schema

SQL_Injection_V3

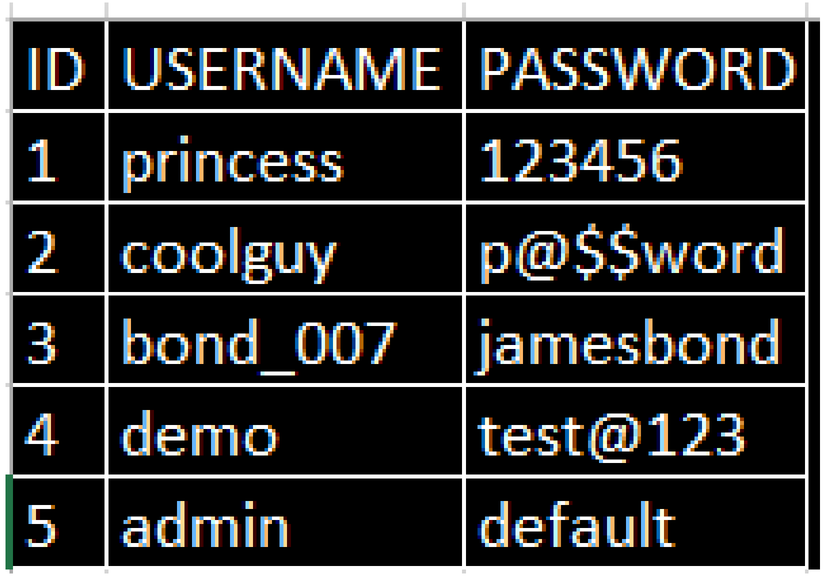
Test

No of tables in SQL_Injection_V3: 2

Hogwarts

Users

Business Impact – Extremely High



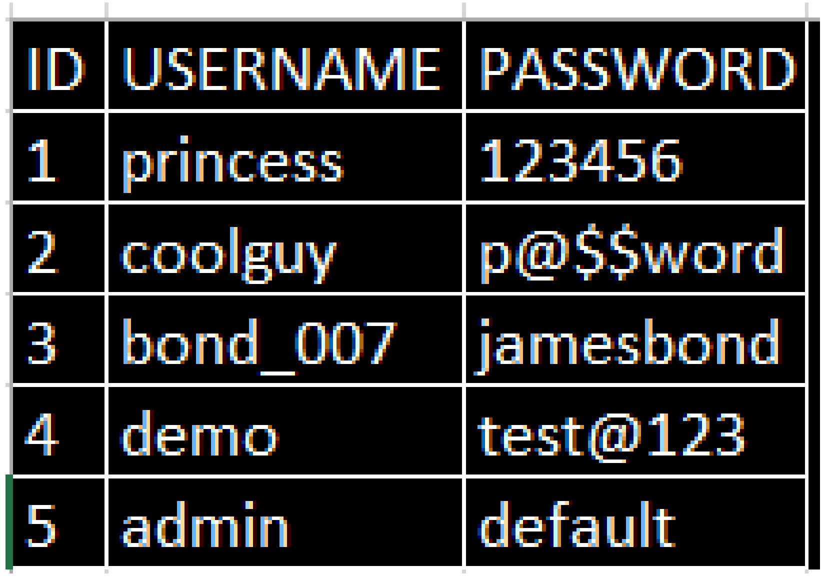
ID	USERNAME	PASSWORD
1	princess	123456
2	coolguy	p@$$word
3	bond_007	jamesbond
4	demo	test@123
5	admin	default

- Using this vulnerability, attacker can execute arbitrary SQL commands on Lifestyle store server and gain complete access to internal databases along with all customer data inside it.
- Below is the screenshot of users table which shows user credentials being leaked 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.

1. SQL Injection

SQL Injection (Critical)	Below mentioned URL in the **Petunia Flowers – Flower Search module** is vulnerable to SQL injection attack **Affected URL :** • http://url.com/petunia/flowerSearch.php **Affected Parameters :** • Flower (POST parameter) **Payload:** • flower=rose’

PoC – Attacker can dump arbitrary data



ID	USERNAME	PASSWORD
1	princess	123456
2	coolguy	p@$$word
3	bond_007	jamesbond
4	demo	test@123
5	admin	default

- No of databases: 3
  - Information_schema
  - SQL_Injection_V3
  - Test
- No of tables in SQL_Injection_V3: 2
  - Hogwarts
  - Users
- Critical Table: Users

Recommendation

- Take the following precautions to avoid exploitation of SQL injections:
  - Whitelist User Input: Whitelist all user input for expected data only. For example if you are expecting a flower name, limit it to alphabets only upto 20 characters in length. If you are expecting some ID, restrict it to numbers only
  - Prepared Statements: Use SQL prepared statements available in all web development languages and frameworks to avoid attacker being able to modify SQL query
  - Character encoding: If you are taking input that requires you to accept special characters, encode it. Example. Convert all **‘ to \’** , **“ to \”**, **\ to \\.** It is also suggested to follow a standard encoding for all special characters such has HTML encoding, URL encoding etc
  - Do not store passwords in plain text. Convert them to hashes using SHA1 SHA256 Blowfish etc
  - Do not run Database Service as admin/root user
  - Disable/remove default accounts, passwords and databases
  - Assign each Database user only the required permissions and not all permissions

References

https://www.owasp.org/index.php/SQL_Injection

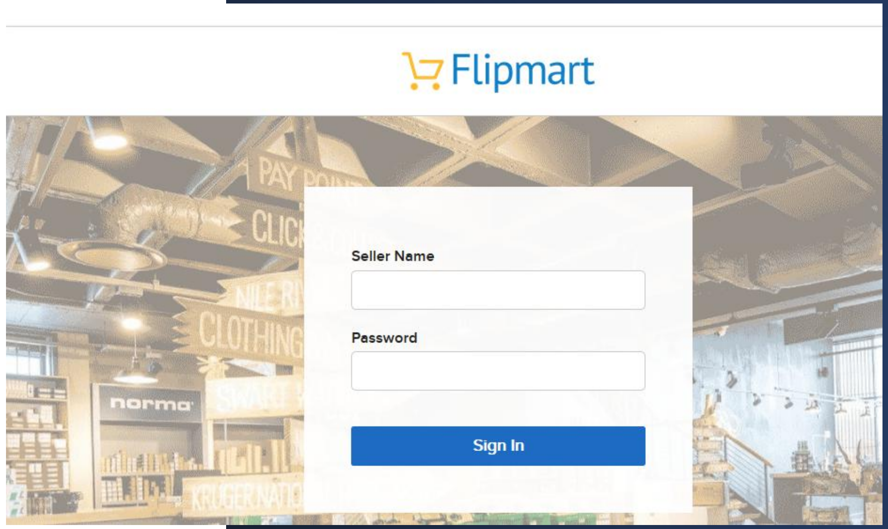
https://en.wikipedia.org/wiki/SQL_injection

2. Access to Sales Dashboard

Access to Sales Dashboard (Critical)	The Sales dashboard at the below mentioned URL has default/weak password allowing complete admin access **Affected URL :** • http://url.com/salesdashboard.php **Affected Parameters :** • Username, password (POST parameters) **Payload:** • Username=admin password=sales@123

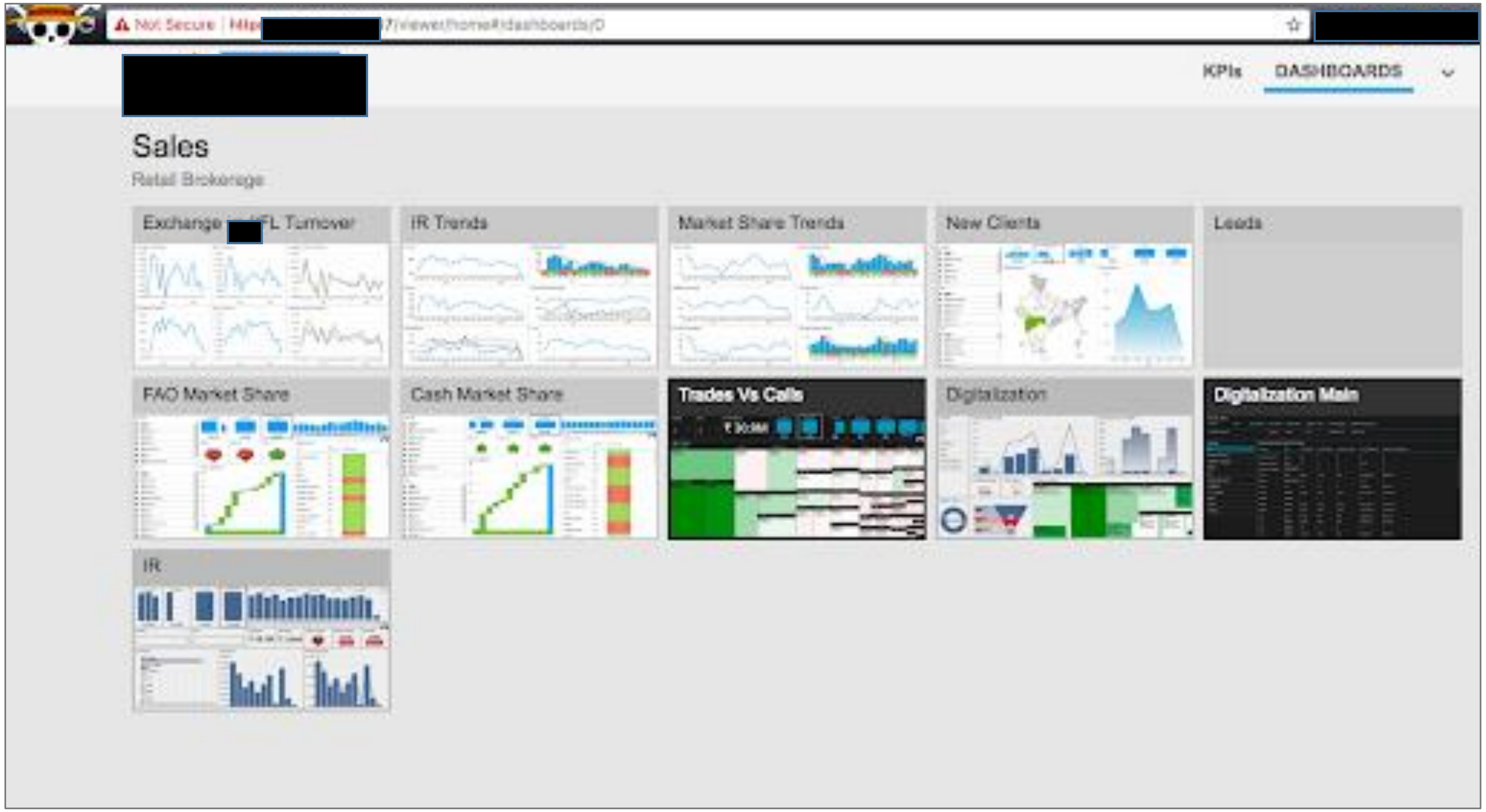
Observation

- Navigate to http://url.com/salesdashboard.php You will see sales admin login page



Observation

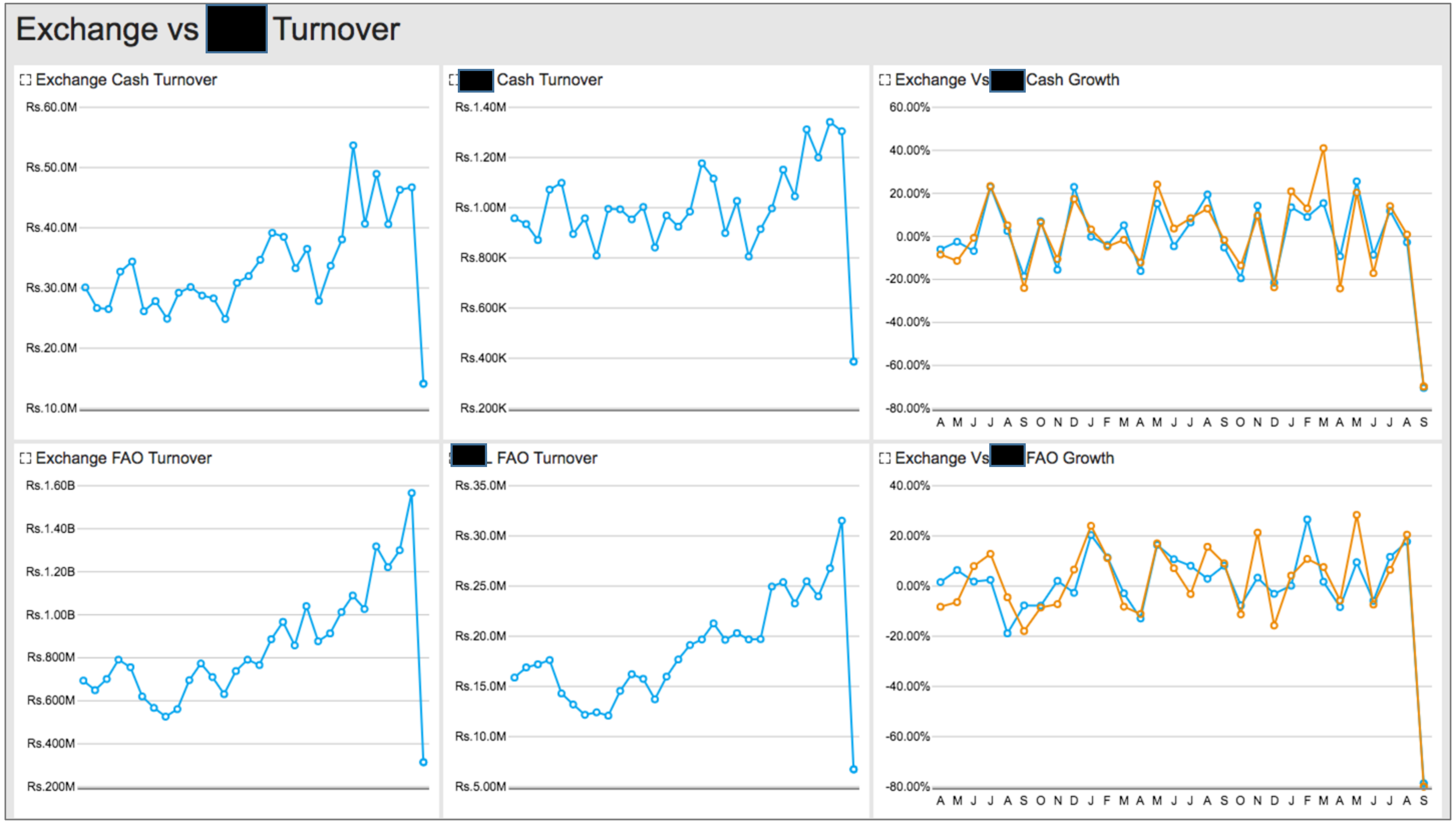
- Enter username: admin & password: sales@123. You will get logged in to the admin panel



Business Impact – Extremely High

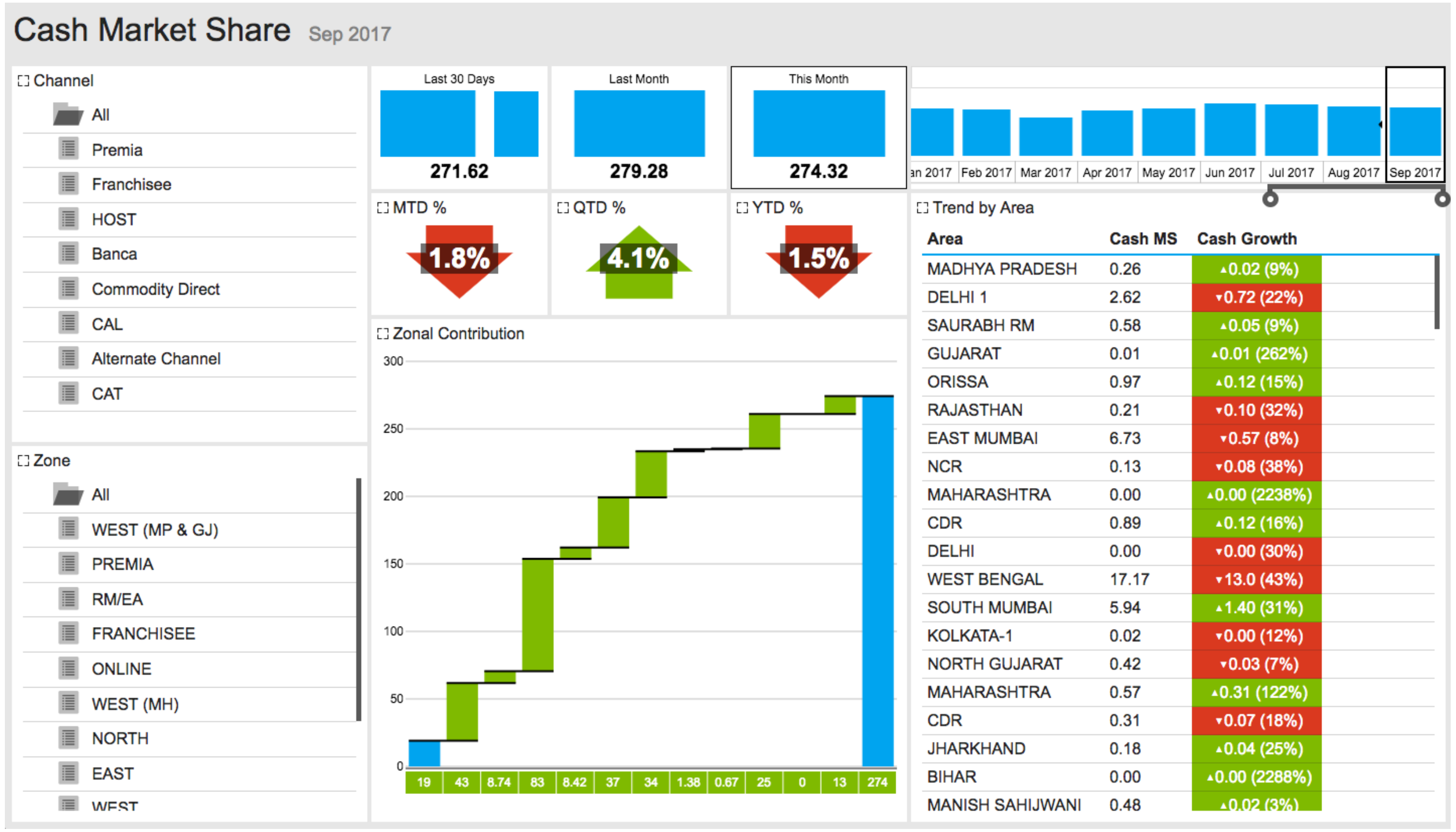
- A malicious user can access the Sales Dashboard which discloses many critical
- information of organization including:
    - Sales Trends
    - Client information
    - Leads information
    - Sales Calendar information
    - Income and revenue information
    - And much more…

POC



Exchange vs [redacted] Turnover
Exchange Cash Turnover
Rs.60.0M
Rs.50.0M
Rs.40.0M
Rs.30.0M
Rs.20.0M
Rs.10.0M
[redacted] Cash Turnover
Rs.1.40M
Rs.1.20M
Rs.1.00M
Rs.800K
Rs.600K
Rs.400K
Rs.200K
Exchange Vs [redacted] Cash Growth
60.00%
40.00%
20.00%
0.00%
-20.00%
-40.00%
-60.00%
-80.00%
A M J J A S O N D J F M A M J J A S O N D J F M A M J J A S
Exchange FAO Turnover
Rs.1.60B
Rs.1.40B
Rs.1.20B
Rs.1.00B
Rs.800M
Rs.600M
Rs.400M
Rs.200M
[redacted] FAO Turnover
Rs.35.0M
Rs.30.0M
Rs.25.0M
Rs.20.0M
Rs.15.0M
Rs.10.0M
Rs.5.00M
Exchange Vs [redacted] FAO Growth
40.00%
20.00%
0.00%
-20.00%
-40.00%
-60.00%
-80.00%
A M J J A S O N D J F M A M J J A S O N D J F M A M J J A S

POC



Cash Market Share Sep 2017

Channel

- All
- Premia
- Franchisee
- HOST
- Banca
- Commodity Direct
- CAL
- Alternate Channel
- CAT

Zone

- All
- WEST (MP & GJ)
- PREMIA
- RM/EA
- FRANCHISEE
- ONLINE
- WEST (MH)
- NORTH
- EAST
- WEST

Last 30 Days	Last Month	This Month
271.62	279.28	274.32

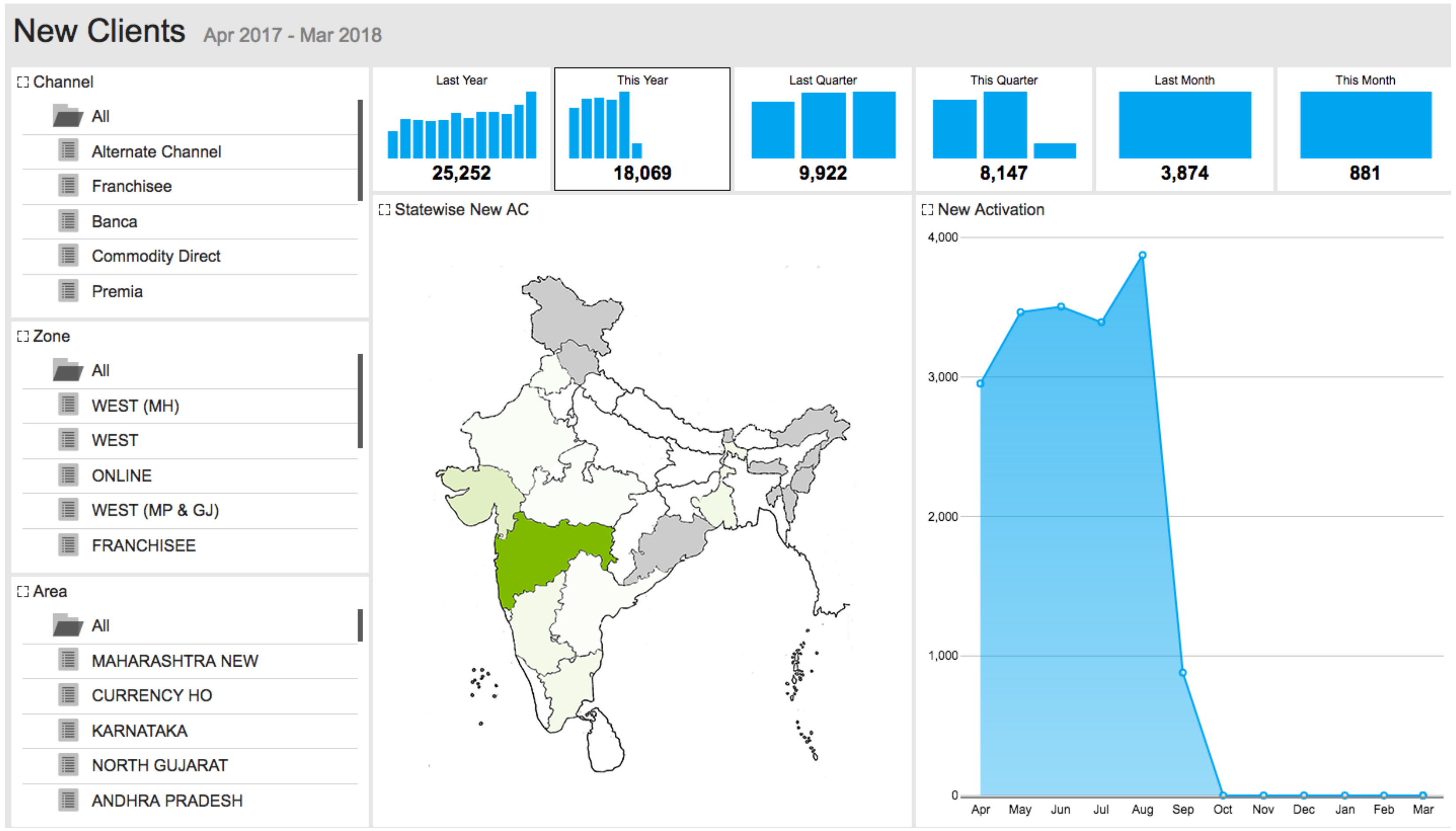
MTD %	QTD %	YTD %
1.8%	4.1%	1.5%

Zonal Contribution

Trend by Area

Area	Cash MS	Cash Growth
MADHYA PRADESH	0.26	▴0.02 (9%)
DELHI 1	2.62	▾0.72 (22%)
SAURABH RM	0.58	▴0.05 (9%)
GUJARAT	0.01	▴0.01 (262%)
ORISSA	0.97	▴0.12 (15%)
RAJASTHAN	0.21	▾0.10 (32%)
EAST MUMBAI	6.73	▾0.57 (8%)
NCR	0.13	▾0.08 (38%)
MAHARASHTRA	0.00	▴0.00 (2238%)
CDR	0.89	▴0.12 (16%)
DELHI	0.00	▾0.00 (30%)
WEST BENGAL	17.17	▾13.0 (43%)
SOUTH MUMBAI	5.94	▴1.40 (31%)
KOLKATA-1	0.02	▾0.00 (12%)
NORTH GUJARAT	0.42	▾0.03 (7%)
MAHARASHTRA	0.57	▴0.31 (122%)
CDR	0.31	▾0.07 (18%)
JHARKHAND	0.18	▴0.04 (25%)
BIHAR	0.00	▴0.00 (2288%)
MANISH SAHIJWANI	0.48	▴0.02 (3%)

POC



New Clients Apr 2017 - Mar 2018

Channel
- All
- Alternate Channel
- Franchisee
- Banca
- Commodity Direct
- Premia

Zone
- All
- WEST (MH)
- WEST
- ONLINE
- WEST (MP & GJ)
- FRANCHISEE

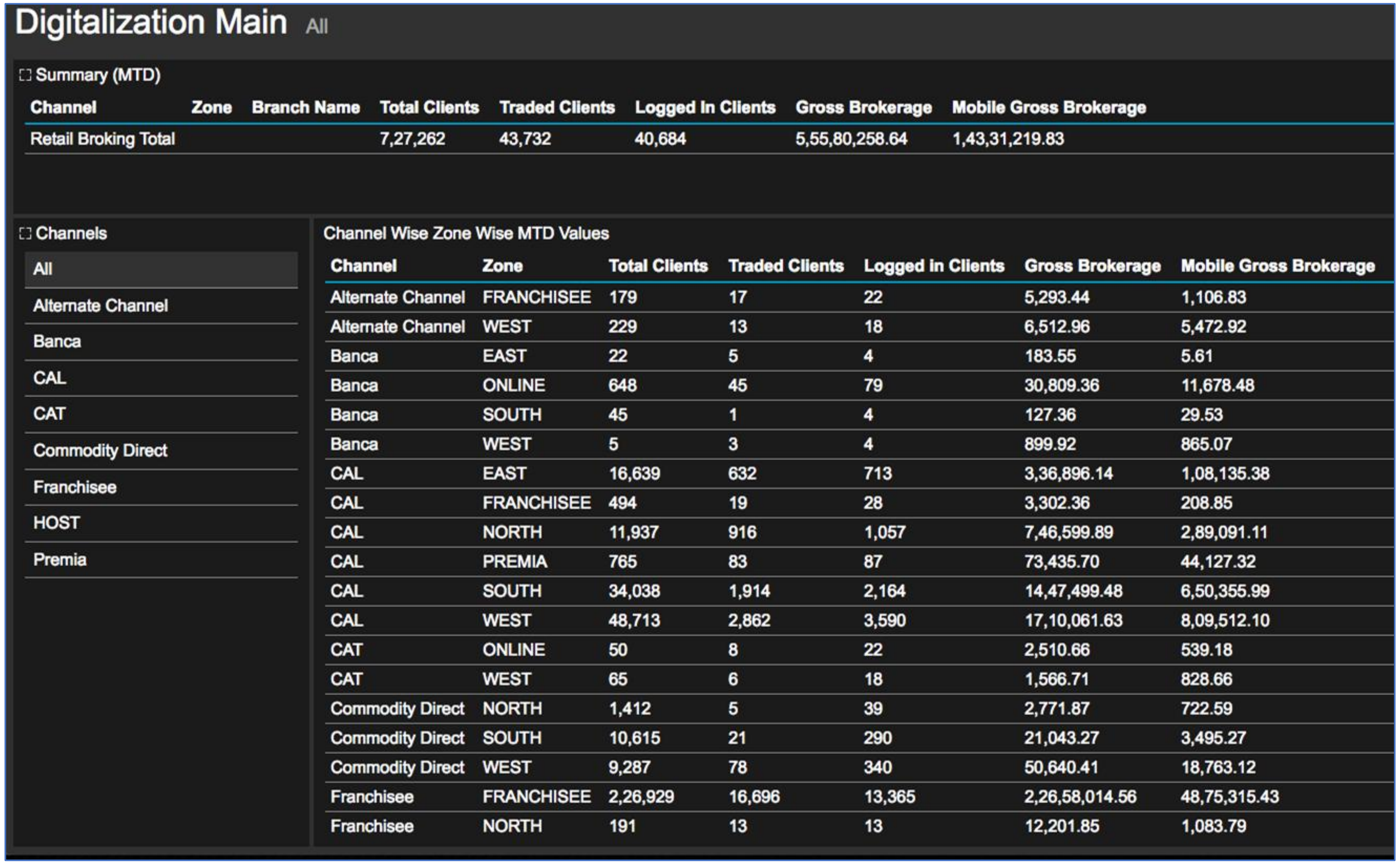
Area
- All
- MAHARASHTRA NEW
- CURRENCY HO
- KARNATAKA
- NORTH GUJARAT
- ANDHRA PRADESH

Last Year	This Year	Last Quarter	This Quarter	Last Month	This Month
25,252	18,069	9,922	8,147	3,874	881

Statewise New AC

New Activation

POC



Digitalization Main All

Summary (MTD)

Channel	Zone	Branch Name	Total Clients	Traded Clients	Logged In Clients	Gross Brokerage	Mobile Gross Brokerage
Retail Broking Total			7,27,262	43,732	40,684	5,55,80,258.64	1,43,31,219.83

Channels

All
Alternate Channel
Banca
CAL
CAT
Commodity Direct
Franchisee
HOST
Premia

Channel Wise Zone Wise MTD Values

Channel	Zone	Total Clients	Traded Clients	Logged In Clients	Gross Brokerage	Mobile Gross Brokerage
Alternate Channel	FRANCHISEE	179	17	22	5,293.44	1,106.83
Alternate Channel	WEST	229	13	18	6,512.96	5,472.92
Banca	EAST	22	5	4	183.55	5.61
Banca	ONLINE	648	45	79	30,809.36	11,678.48
Banca	SOUTH	45	1	4	127.36	29.53
Banca	WEST	5	3	4	899.92	865.07
CAL	EAST	16,639	632	713	3,36,896.14	1,08,135.38
CAL	FRANCHISEE	494	19	28	3,302.36	208.85
CAL	NORTH	11,937	916	1,057	7,46,599.89	2,89,091.11
CAL	PREMIA	765	83	87	73,435.70	44,127.32
CAL	SOUTH	34,038	1,914	2,164	14,47,499.48	6,50,355.99
CAL	WEST	48,713	2,862	3,590	17,10,061.63	8,09,512.10
CAT	ONLINE	50	8	22	2,510.66	539.18
CAT	WEST	65	6	18	1,566.71	828.66
Commodity Direct	NORTH	1,412	5	39	2,771.87	722.59
Commodity Direct	SOUTH	10,615	21	290	21,043.27	3,495.27
Commodity Direct	WEST	9,287	78	340	50,640.41	18,763.12
Franchisee	FRANCHISEE	2,26,929	16,696	13,365	2,26,58,014.56	48,75,315.43
Franchisee	NORTH	191	13	13	12,201.85	1,083.79

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/Default_Passwords
https://www.us-cert.gov/ncas/alerts/TA13-175A

3. Account Takeover Using OTP Bypass

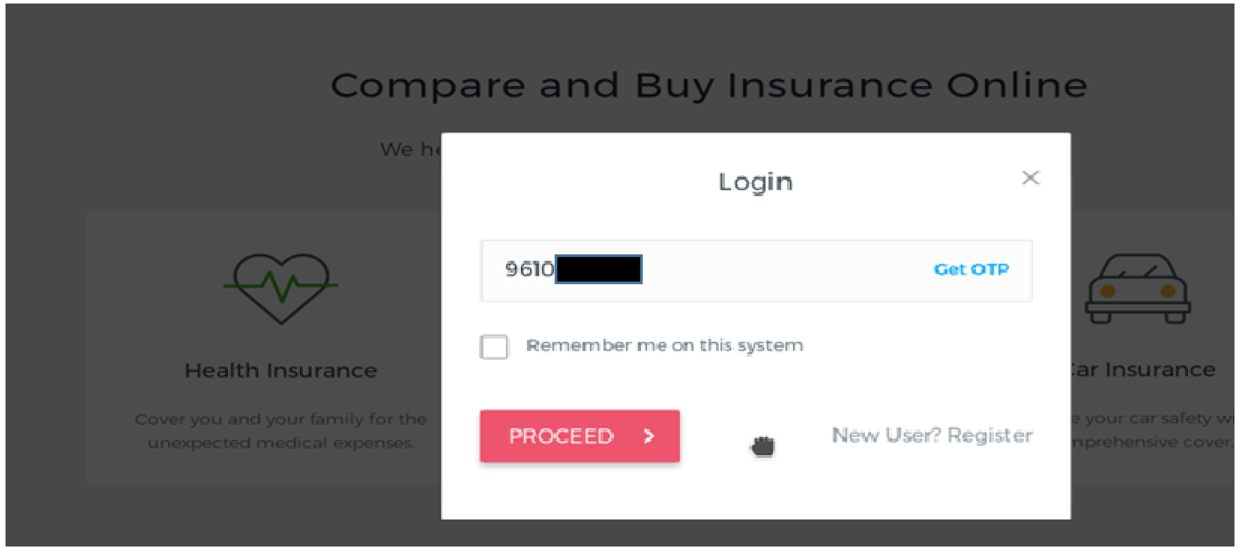
Account Takeover Using OTP Bypass (Critical)	The below mentioned login page allows login via OTP which can be bruteforced **Affected URL :** • http://url.com/login_via_OTP.php **Affected Parameters :** • OTP (POST parameters)

3. Account Takeover Using OTP Bypass

Account Takeover Using OTP Bypass (Critical)	Similar issue is observed on the below mentioned login pages too **Affected URL :** • http://url.com/admin/login_via_OTP.php **Affected Parameters :** • code (POST parameters)

Observation

- Navigate to http://url.com/login_via_OTP.php You will see user login page via OTP. Enter victim's mobile number while capturing requests in a local proxy and click Get OTP


Compare and Buy Insurance Online
We h
Health Insurance
Cover you and your family for the unexpected medical expenses.
ar Insurance
e your car safety w
nprehensive cover.

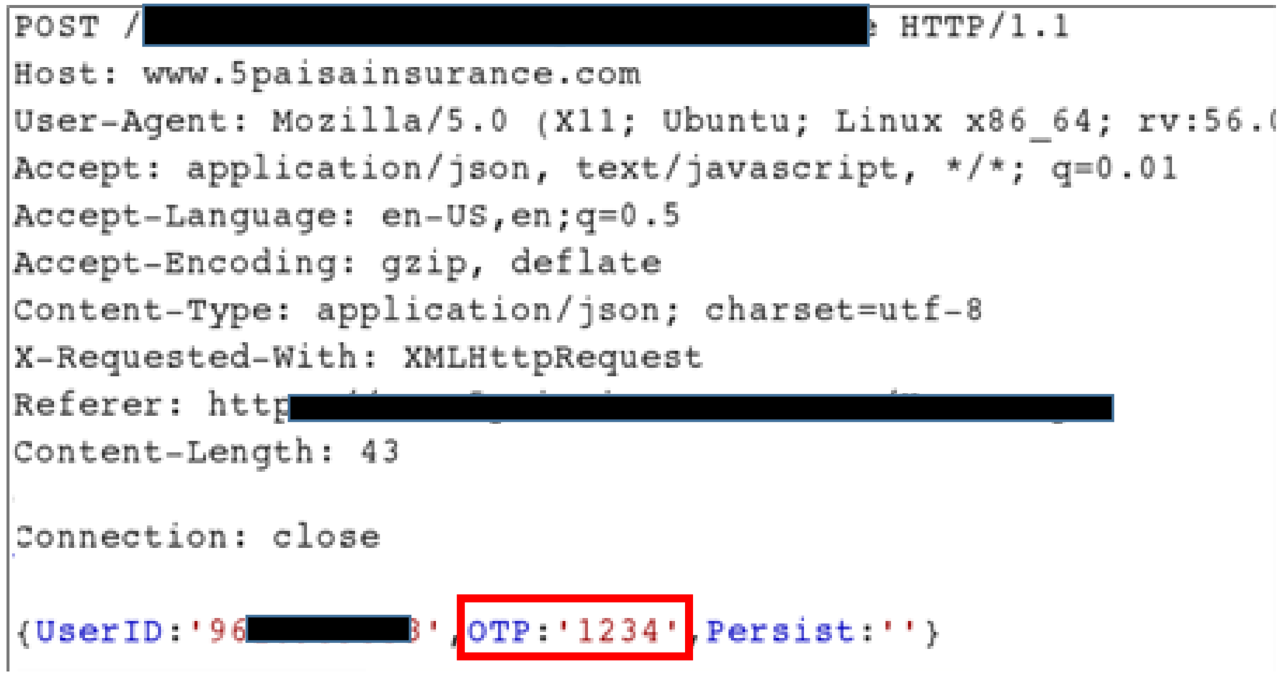
Login

9610[redacted]

Get OTP
Remember me on this system
PROCEED
New User? Register

Observation

- Following request will be generated containing OTP parameter.



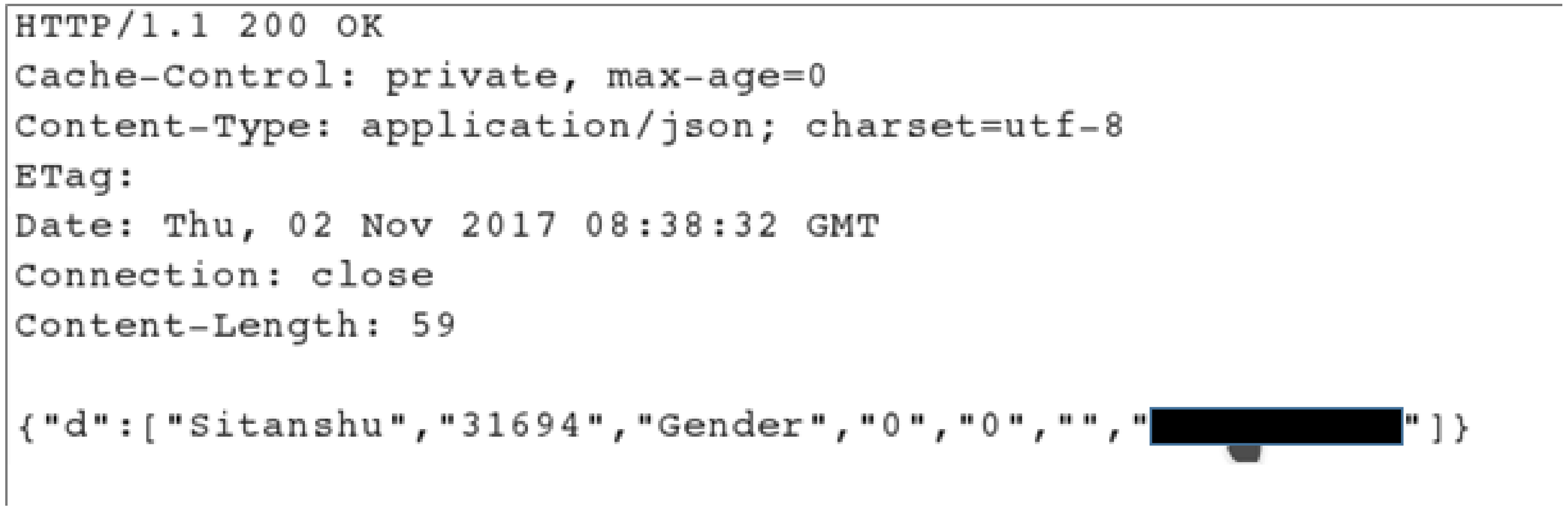
```
POST /[redacted] HTTP/1.1
Host: www.5paisainsurance.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:56.
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
X-Requested-With: XMLHttpRequest
Referer: http[redacted]
Content-Length: 43

Connection: close

{UserID:'96[redacted]',OTP:'1234',Persist:''}
```

Observation

- We shoot the request with all possible combinations of 4 Digit OTPs and upon a successful hit, we get a response containing user details. We can use the same OTP then to login.

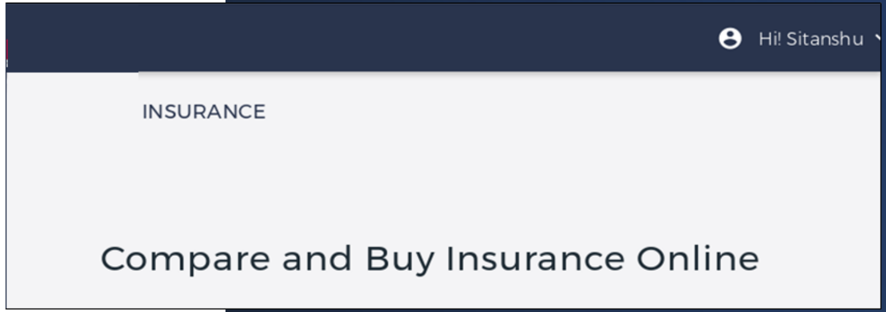


```
HTTP/1.1 200 OK
Cache-Control: private, max-age=0
Content-Type: application/json; charset=utf-8
ETag:
Date: Thu, 02 Nov 2017 08:38:32 GMT
Connection: close
Content-Length: 59

{"d":["Sitanshu","31694","Gender","0","0","","[redacted]"]}
```

Business Impact – Extremely High

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



Recommendation

Take the following precautions:

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

References:

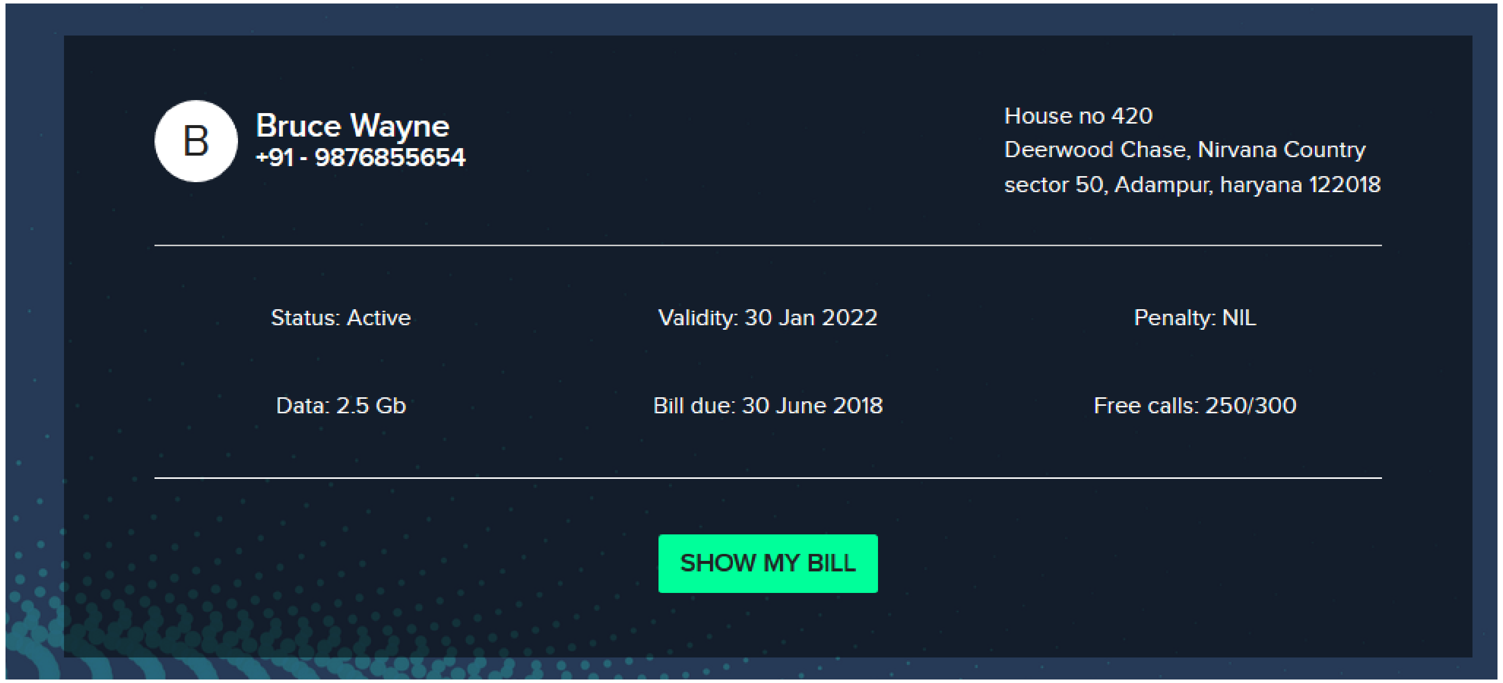
https://www.owasp.org/index.php/Testing_Multiple_Factors_Authentication_(OWASP-AT-009)
https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks

4. Unauthorised Access to Customer Details

Unauthorised Access to Customer Details (Critical)	The Show My Bill module suffers from an Insecure Direct Object Reference (IDOR) that allows attacker get access to anyones Bill details **Affected URL :** • http://hackingenv.internshala.com/Insecure-Direct-Object-Reference/GET-Based-IDOR-in-URL-Variant-1/bill.php **Affected Parameters :** • user_id (GET parameters)

4. Unauthorised Access to Customer Details

Unauthorised Access to Customer Details (Critical)	Similar issue is found on below modules too **Affected URL :** • http://url/invoice.php **Affected Parameters :** • invoice_id (GET parameter) **Affected URL :** • http://url/call_history.php **Affected Parameters :** • mobile_no (POST parameter) **Affected URL :** • http://url/recharge.php **Affected Parameters :** • from_accountno (POST parameter) **Affected URL :** • http://url/sms_history.php **Affected Parameters :** • mobile_no(GET parameter)


B Bruce Wayne
+91 - 9876855654

House no 420
Deerwood Chase, Nirvana Country
sector 50, Adampur, haryana 122018

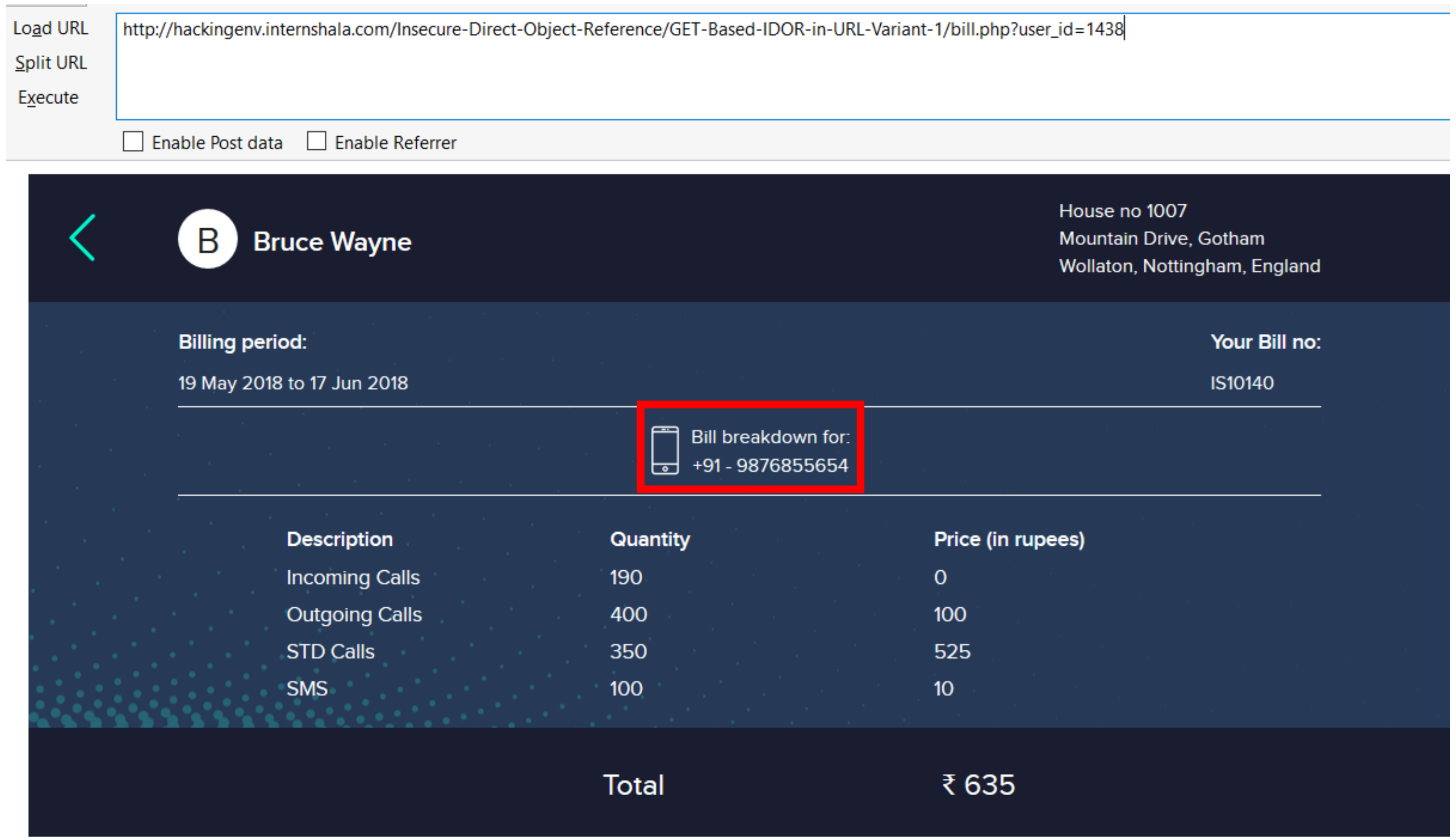
Status: Active	Validity: 30 Jan 2022	Penalty: NIL
Data: 2.5 Gb	Bill due: 30 June 2018	Free calls: 250/300

- Login to your account and navigate to Bill page on http://hackingenv.internshala.com/Insecure-Direct-Object-Reference/GET-Based-IDOR-in-URL-Variant-1/ and click on Show My Bill button

Observation

Observation

- Your bill will be shown to you like below. Notice the URL: http://hackingenv.internshala.com/Insecure-Direct-Object-Reference/GET-Based-IDOR-in-URL-Variant-1/bill.php?user_id=1438
- It contains user_id of our user and we get bill details of our user's **mobile number: 9876855654**



```
http://hackingenv.internshala.com/Insecure-Direct-Object-Reference/GET-Based-IDOR-in-URL-Variant-1/bill.php?user_id=1438
```

B Bruce Wayne

House no 1007
Mountain Drive, Gotham
Wollaton, Nottingham, England

Billing period: 19 May 2018 to 17 Jun 2018

Your Bill no: IS10140

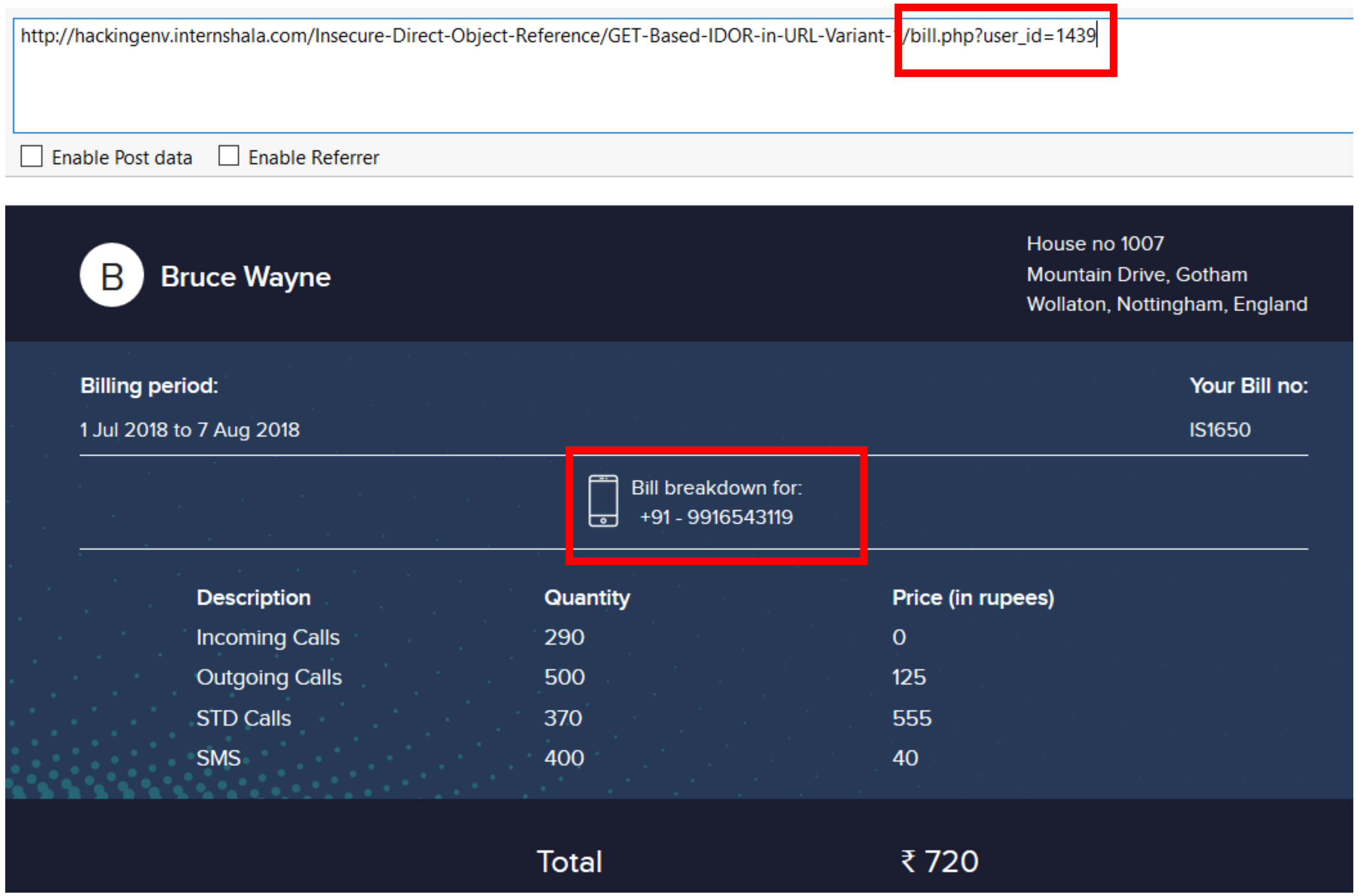
Bill breakdown for: +91 - 9876855654

Description	Quantity	Price (in rupees)
Incoming Calls	190	0
Outgoing Calls	400	100
STD Calls	350	525
SMS	100	10

Total ₹ 635

Observation

- We change this user_id from 1438 to 1439 and we get bill information of a different user with **mobile number: 9976543119**



http://hackingenv.internshala.com/Insecure-Direct-Object-Reference/GET-Based-IDOR-in-URL-Variant-1/bill.php?user_id=1439

☐ Enable Post data ☐ Enable Referrer

Bruce Wayne

House no 1007
Mountain Drive, Gotham
Wollaton, Nottingham, England

Billing period:
1 Jul 2018 to 7 Aug 2018

Your Bill no:
IS1650

Bill breakdown for:
+91 - 9916543119

Description	Quantity	Price (in rupees)
Incoming Calls	290	0
Outgoing Calls	500	125
STD Calls	370	555
SMS	400	40

Total ₹ 720

Business Impact – Extremely High

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

- Mobile Number
- Bill Number
- Billing Period
- Bill Amount and Breakdown

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.

Other IDORs on the application are leaking much more information including Payment details, call history and even allow attacker to recharge his mobile number deducting money from any one else’s account which can be used to steal money from users.

As a PoC, Bill details of 100 users are dumped in the attached excel file below:

Microsoft Excel
7-2003 Workshe

Recommendation

Take the following precautions:

- Implement proper authentication and authorisation 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

5. Reflected Cross Site Scripting (XSS)

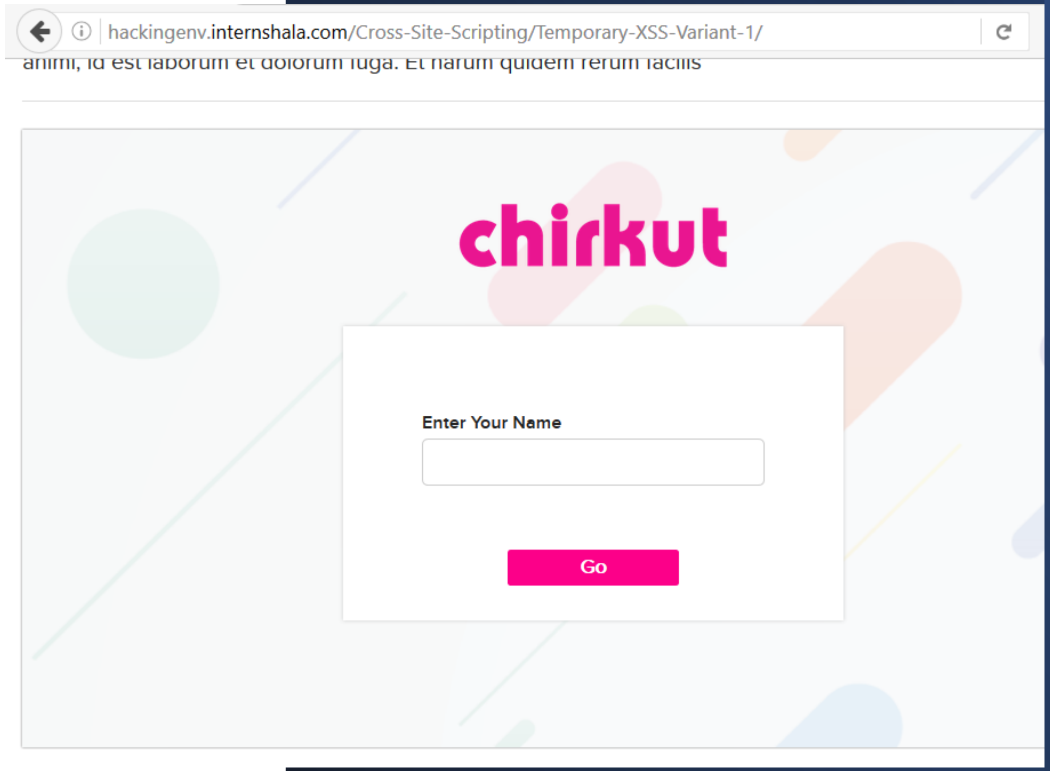
Reflected Cross Site Scripting (Severe)	Below mentioned parameters are vulnerable to reflected XSS **Affected URL :** • hackingenv.internshala.com/Cross-Site-Scripting/Temporary-XSS-Variant-1/hello.php **Affected Parameters :** • user_name(GET parameters) **Payload:** • <script>alert(1)</script>

5. Reflected Cross Site Scripting (XSS)

Reflected Cross Site Scripting (Severe)	Similar issue is found on below modules too **Affected URL :** • http://hackingenv.internshala.com/Cross-Site-Scripting/Temporary-XSS-Variant-2/xss/testing* **Affected Parameters :** • URL – anything after testing **Payload:** • <body onload=alert(1)>
	Affected URL : • http://hackingenv.internshala.com/Cross-Site-Scripting/Temporary-XSS-Variant-4/ **Affected Parameters :** • url (POST parameters) **Payload:** • " onload="alert(1)

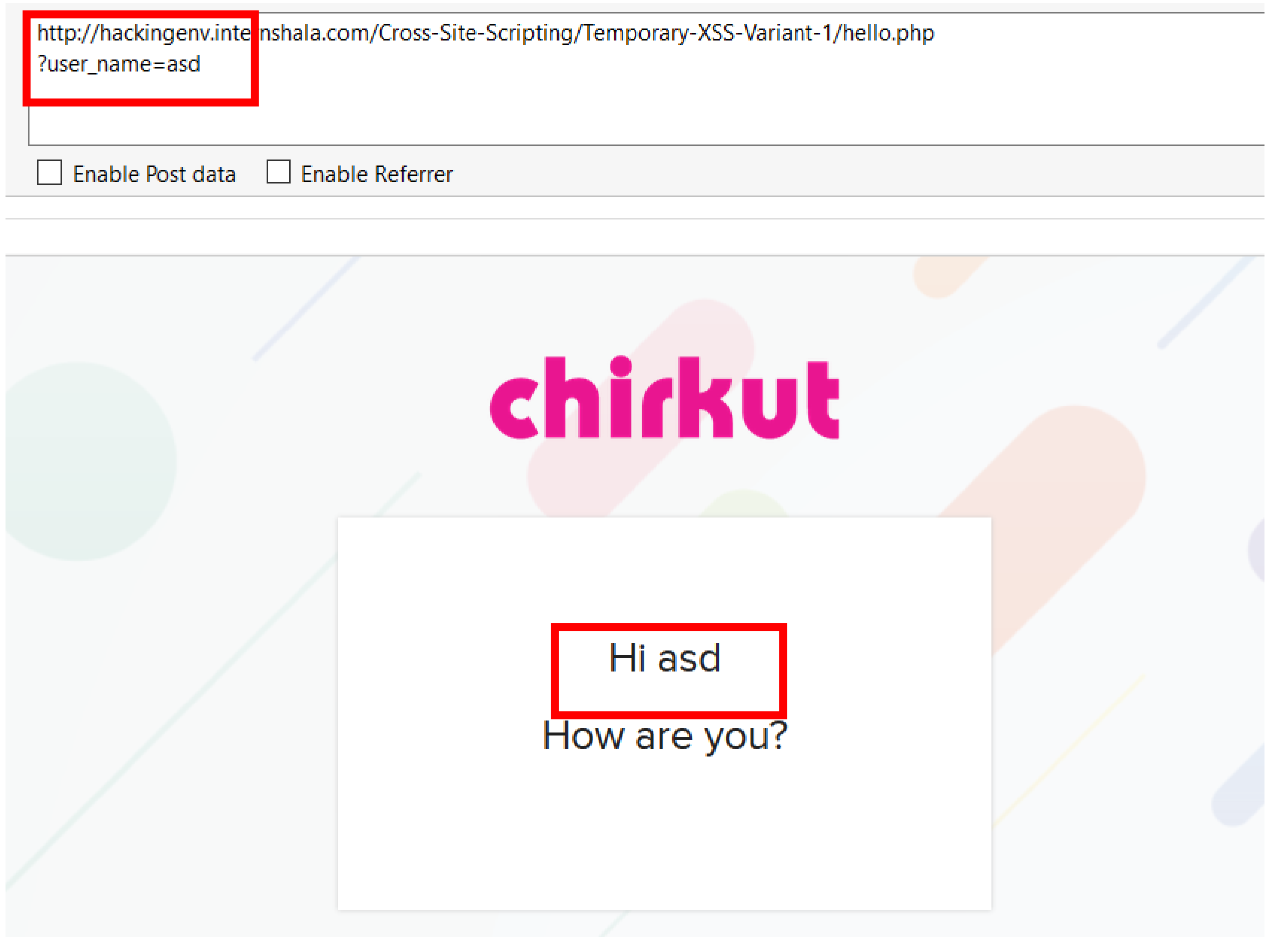
Observation

- Navigate to hackingenv.internshala.com/Cross-Site-Scripting/Temporary-XSS-Variant-1/hello.php
- You will see a field to enter some text


hackingenv.internshala.com/Cross-Site-Scripting/Temporary-XSS-Variant-1/
animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis
chirkut
Enter Your Name
Go

Observation

Enter any text and click the button, you will see it reflected in the next page and value will be in GET parameter **user_name**



http://hackingenv.internshala.com/Cross-Site-Scripting/Temporary-XSS-Variant-1/hello.php
?user_name=asd

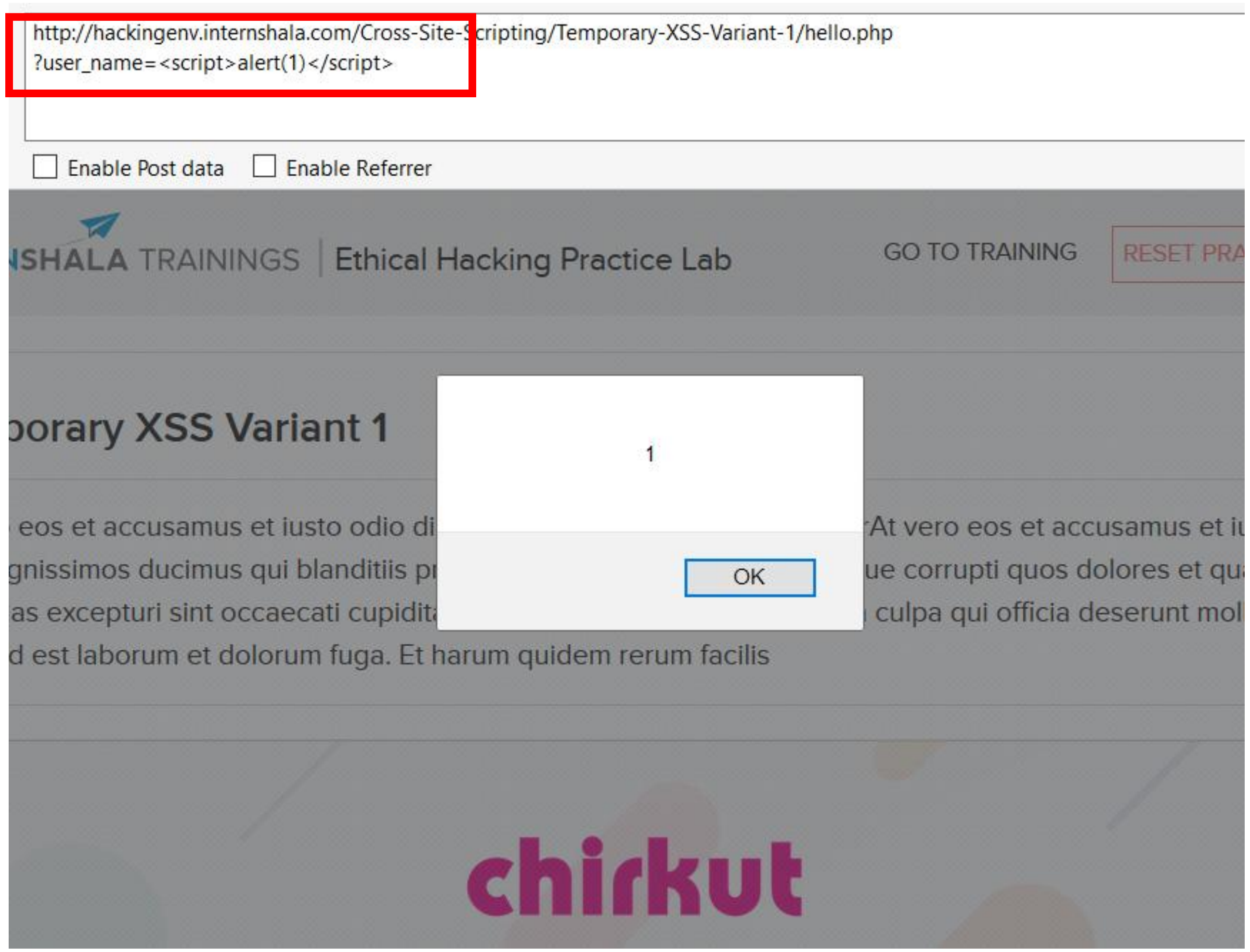
Hi asd

How are you?

Observation

Put the payload instead of asd: **<script>alert(1)</script>**

As you can see we executed custom JS causing popup

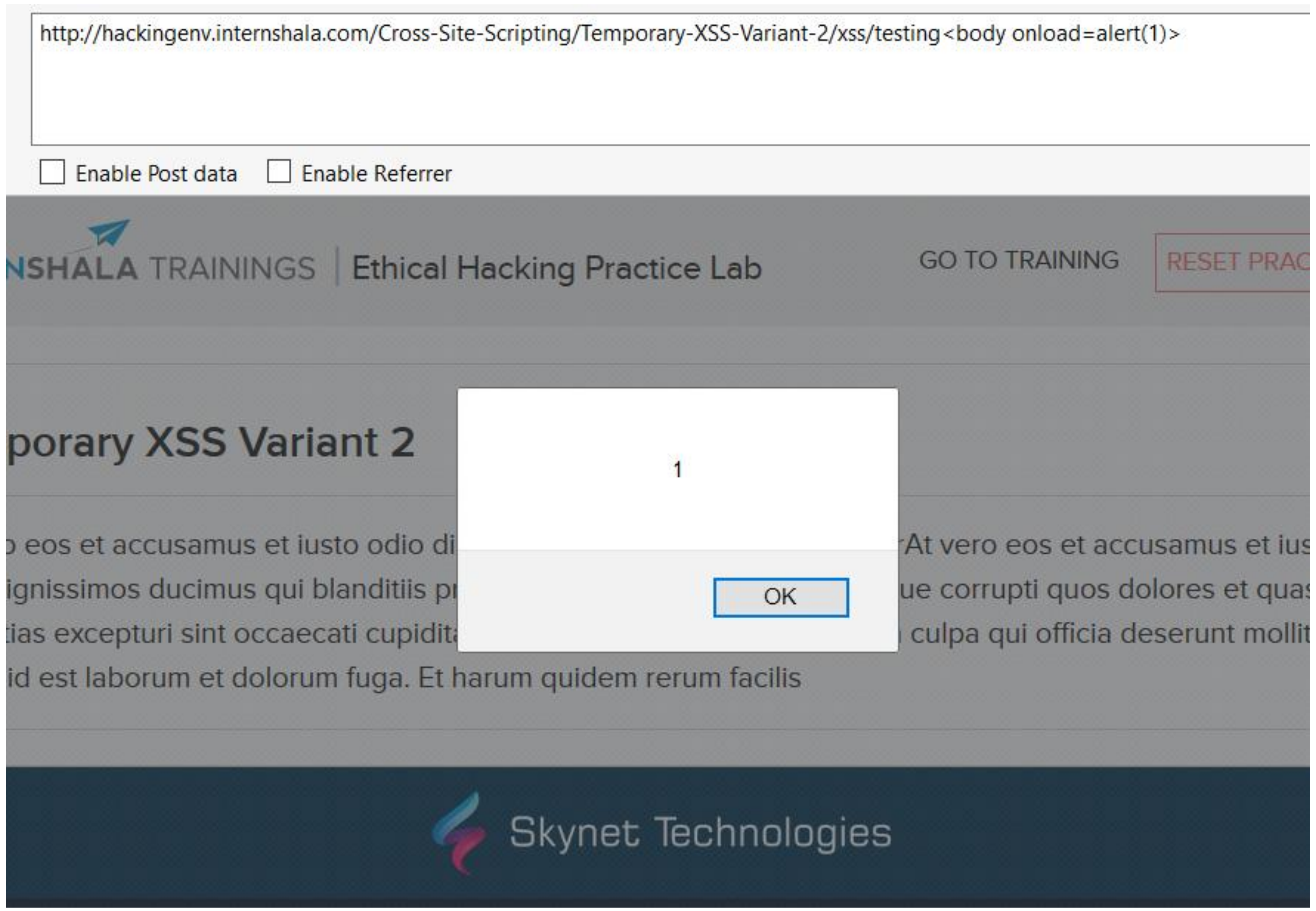


```
http://hackingenv.internshala.com/Cross-Site-Scripting/Temporary-XSS-Variant-1/hello.php
?user_name=<script>alert(1)</script>
```

eos et accusamus et iusto odio di
gnissimos ducimus qui blanditiis pr
as excepturi sint occaecati cupidit
d est laborum et dolorum fuga. Et harum quidem rerum facilis

At vero eos et accusamus et iu
ue corrupti quos dolores et qu
culpa qui officia deserunt mol

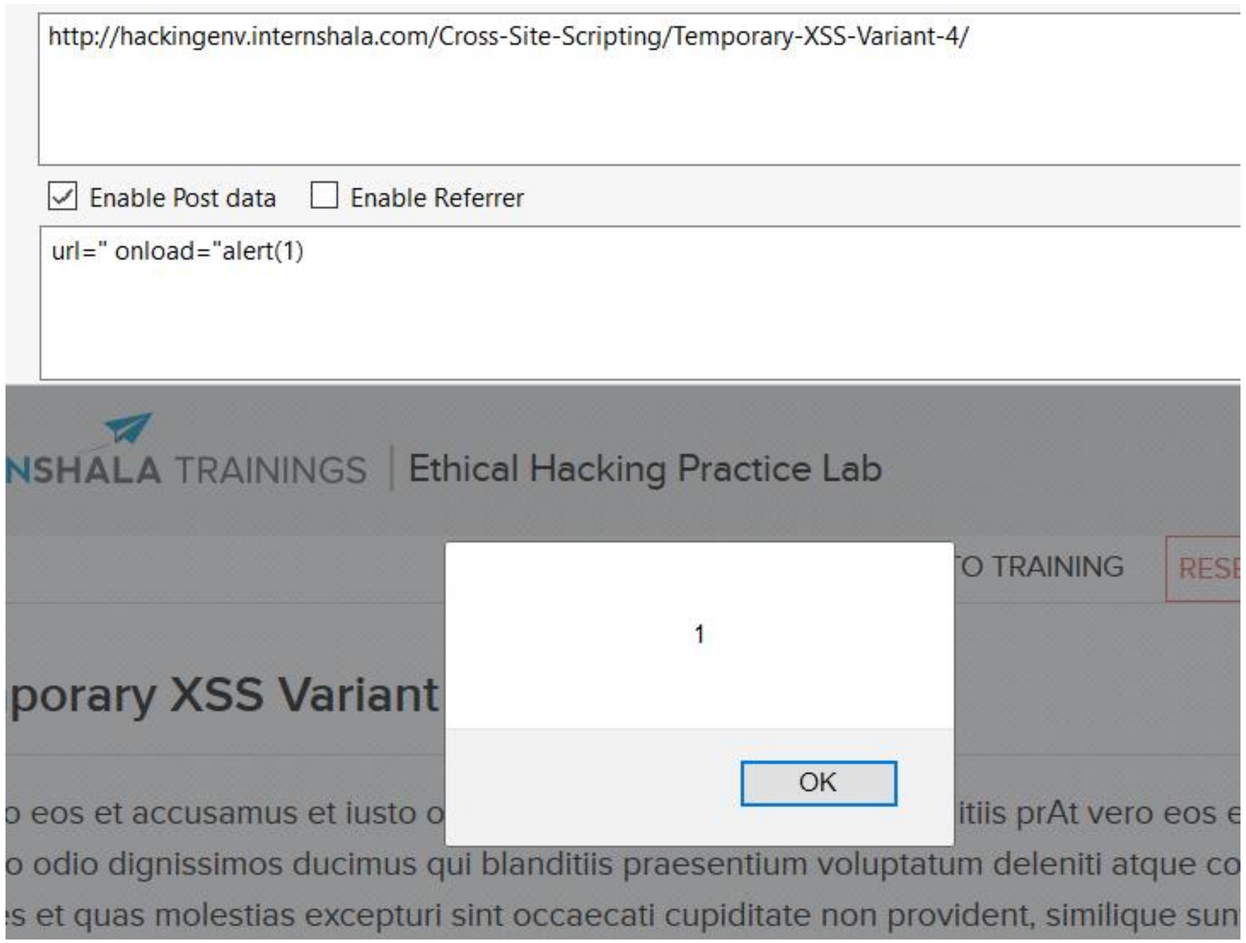
PoC



```
http://hackingenv.internshala.com/Cross-Site-Scripting/Temporary-XSS-Variant-2/xss/testing<body onload=alert(1)>
```

eos et accusamus et iusto odio di ignissimos ducimus qui blanditiis pr ias excepturi sint occaecati cupidit id est laborum et dolorum fuga. Et harum quidem rerum facilis At vero eos et accusamus et ius ue corrupti quos dolores et quas culpa qui officia deserunt mollit

PoC



```
http://hackingenv.internshala.com/Cross-Site-Scripting/Temporary-XSS-Variant-4/
```

☑ Enable Post data ☐ Enable Referrer

```
url=" onload="alert(1)
```

porary XSS Variant

o eos et accusamus et iusto o itiis prAt vero eos e
o odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque co
s et quas molestias excepturi sint occaecati cupiditate non provident, similique sun

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:

- Sanitise all user input and block characters you do not want
- Convert special HTML characters like ‘ “ < > into HTML entities &quot; %22 &lt; &gt; before printing them on the website

References:

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

6. Directory Listing

Directory Listing (Moderate)	Below mentioned parameters are vulnerable to reflected XSS **Affected URL :** • http://URL1/backup/ • http://url2/profile_pictures/

Observation

- Navigate to http://URL1/backup/
- Complete listing of directory is shown containing month wise HTML backups of the website

Observation

- Navigate to http://URL2/profile_pictures/
- Complete listing of directory is shown containing profile pictures of all users on the website

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

- Also, attacker can simply download the backups and images and view them

Recommendation

Take the following precautions:

- Disable Directory Listing
- Put an index.html in all folders with default message

References:

https://cwe.mitre.org/data/definitions/548.html
https://www.netsparker.com/blog/web-security/disable-directory-listing-web-servers/

7. Information Disclosure

Information Disclosure due to Apache Info Pages (Low)	Below mentioned urls disclose server information **Affected URL :** • http://URL/server-status • http://URL/server-info

Observation

- Navigate to mentioned URL
- Default server-status page opens which discloses server information

Observation

- server-info page

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

Recommendation

Take the following precautions:

- Disable all default pages and folders including server-status and server-info

References:

https://vuldb.com/?id.88482
https://httpd.apache.org/docs/current/mod/mod_status.html
https://www.beyondsecurity.com/scan_pentest_network_vulnerabilities_apache_http_server_httponly_cookie_information_disclosure

THANK YOU

For any further clarifications/patch assistance, please contact: 9765310126