

# Summer Internship Report

On

**ICSI|CNSS Certified Network Security Specialist**

**Submitted in partial fulfillment of the  
requirement for the award  
of the degree of**

**BACHLOR OF TECHNOLOGY  
INFORMATION TECHNOLOGY  
(2018-2022)**



**Submitted By:- Hitesh Kumar Meena  
(01520703118)**



# **CERTIFICATE OF COMPLETION**

**HITESH KUMAR MEENA**

*has successfully completed the course*

**ICSI | CNSS Certified Network  
Security Specialist**

May 18, 2020

A handwritten signature in black ink, reading 'Gthrasyvoulou', is positioned above the printed name of the Program Director.

**George Thrasyvoulou**  
Program Director

Credential Number: 18265411

## CANDIDATE'S DECLARATION

I **Hitesh Kumar Meena** hereby declare that I have undertaken six/four weeks industrial training at **ICSI** during a period from **2 April 2020 to 18 May 2020** in partial fulfillment of requirements for the award of degree of B.Tech (INFORMATION TECHNOLOGY) At **Netaji Subhas University of Technology (West Campus), DELHI**. The work which is being presented in the training report submitted to Department of INFORMATION TECHNOLOGY At **Netaji Subhas University of Technology (West Campus)**, is an authentic record of training work.

Signature of the Student

The six/four weeks industrial training Viva–Voce Examination of \_\_\_\_\_ has been held on \_\_\_\_\_ \_ and accepted.

Signature of Internal Examiner

Signature of External Examiner

# Abstract

**Cybersecurity** for data networks is in its infancy while attackers on networks are becoming increasingly sophisticated. The necessary widespread use of wireless networks provides more vulnerabilities. Network routing is key to a functioning network; once compromised, it can be difficult to recover. For many years, network practitioners have worked on methods to protect that routing through authentication of the updates passed in the network. The missing piece has been a usable, protectable key management system. This proposal uses recent advances in the creation of locally controlled and administered hierarchical web-of-trust certificates to provide a managed secure identity for routers that can be used to protect network routes from attack and misconfiguration.

This proposal is the first phase of creating an authenticated routing infrastructure. The work involves adapting advances in evidentiary trust to a link state routing protocol, developing naming for certificate chains and an approach to use the certificates in link state updates. This phase is expected to result in a report and a design for a prototype to be added to open source

protocols in a later phase. These results will be made publicly available through open source code and discussions and presentations at standards bodies and with router vendors and network operators. A successful approach should create opportunities to the proposer in contracts with government and commercial organizations. This could result in market opportunities for other organizations such as network management tool and router vendors.

The use of evidentiary trust is expected to have other applications, but its application to the routing infrastructure can have nearer- term impact on securing networks important to the government.

## **Acknowledgement**

I take this opportunity to express my profound gratitude and deep regards to **ICSI** for their exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by them time to time shall carry me a long way in the journey of life on which I am about to embark.

I take this opportunity to thank **Netaji Subhas University of Technology (West Campus)** for giving me chance to do this project.

Lastly, I would like to thank each and every person who directly or indirectly helped me in the completion of the project especially my parents and Peers who supported me throughout my project.



## **INTRODUCTION**

The UK registered and accredited International CyberSecurity Institute (ICSI) was created in response to two distinct demands in the market. On the one hand, cybersecurity is an important issue of growing concern around the world. And yet, the shortage of qualified professionals is widely acknowledged, as the industry struggles with ever-increasing breaches in security. On the other hand, academic qualifications provided by universities are often more focused on theoretical knowledge rather than practical, hands-on training.

ICSI's strength lies in the accredited courses that are delivered by practicing and specialized experts in Cybersecurity. ICSI's core programmes have earned accreditation from CREST and NCSC, two independent UK bodies that are recognized around the world. Furthermore, our courses are accredited by the University of Central Lancashire (UCLAN), UK.

ICSI (International CyberSecurity Institute) is based in Milton Keynes, 80km northwest of London. The International CyberSecurity Institute (ICSI) is a UK registered and accredited, ISO 27001 certified cybersecurity education firm.

### **In association with:**

- 1) CREST Approved Training Provider**
- 2) NCSC Certified Training**
- 3) ISO 27001 Certification**
- 4) University of Central Lancashire**

<b>CHAPTER NO.</b>	<b>CONTENT</b>	<b>PAGE NO.</b>
	<b>CERTIFICATE</b>	II
	<b>DECLARATION</b>	III
	<b>ABSTRACT</b>	IV
	<b>ACKNOWLEDGEMENT</b>	V
	<b>INTRODUCTION OF ICSI</b>	iv
	<b>CHAPTER 1 : INTRODUCITON TO ETHICAL HACKING</b>	1-7
	1.1 INTRODUCTION AND JOB OF AN ETHICAL HACKER	1
	1.2 TYPES OF HACKERS	2
	1.3 TYPES OF HACKING	3
	1.4 TYPES OF WEB SECURITY VULNERABILITIES	4
	1.5 THE 6 PHASES OF ETHICAL HACKING	7
	<b>CHAPTER 2 : PROBLEM STATEMENT</b>	8-31
	<b>CHAPTER 3 : RESULT &amp; DISCUSSION</b>	32-33
	<b>CHAPTER 4 : CONCLUSION AND RECOMMENDATION</b>	34
	<b>REFERENCES</b>	35

# CHAPTER-1

## INTRODUCITON TO ETHICAL HACKING

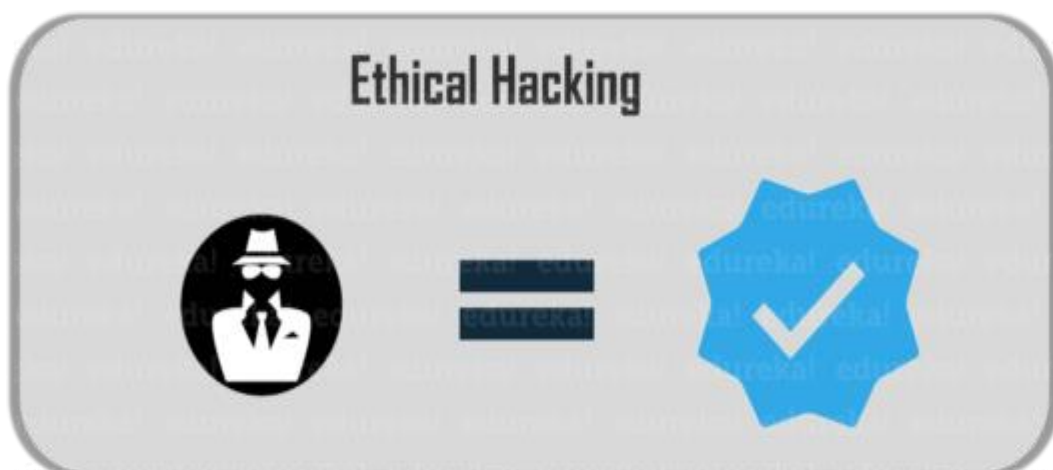
### 1.1 INTRODUCTION AND JOB OF AN ETHICAL HACKER

#### What is Ethical Hacking?

Hacking is the process of finding vulnerabilities in a system and using these found vulnerabilities to gain unauthorized access into the system to perform malicious activities ranging from deleting system files to stealing sensitive information. Hacking is illegal and can lead to extreme consequences if you are caught in the act. People have been sentenced to years of imprisonment because of hacking.



Nonetheless, hacking can be legal if done with permission. Computer experts are often hired by companies to hack into their system to find vulnerabilities and weak endpoints so that they can be fixed. This is done as a precautionary measure against legitimate hackers who have malicious intent. Such people, who hack into a system with permission, without any malicious intent, are known as *ethical hackers* and the process is known as *ethical hacking*.





## 1.2 TYPES OF HACKERS

Hackers can be segregated according to their intent.

### White Hat Hacker



it is another name for an Ethical Hacker. They hack into a system with prior permission to find out vulnerabilities so that they can be fixed before a person with malicious intent finds them.

### Black Hat Hacker



They are also known as crackers, who hack in order to gain unauthorized access to a system & harm its operations or steal sensitive information. It's always illegal because of its malicious intent which includes stealing corporate data, violating privacy, damaging the system etc.

### Grey Hat Hacker



They are a blend of both black hat and white hat hackers. They mostly hack for fun and exploit a security weakness in a computer system or network without the owner's permission or knowledge. Their intent is to bring the weakness to the attention of the owners & earning some bug bounty.

### Suicide Hacker



A suicide hacker is a person who works with the intent to bring down major corporations and infrastructure. These kinds of hackers are not scared of the consequences of their actions as they mostly work with a vengeance in their mind. These people are also known as hacktivists.

## 1.3 TYPES OF HACKING

We can segregate hacking into different types depending on what the hacker is trying to achieve.

### Website Hacking



Hacking a website means taking unauthorized control over a web server and its associated software such as databases and other interfaces.

### Network Hacking



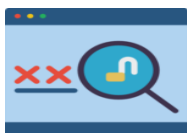
Hacking a network means gathering information about a network by using tools like Telnet, NS lookup, Ping, Tracert, Netstat, etc. with the intent to harm the network system and hamper its operation.

### Email Hacking



This includes gaining unauthorized access to an Email account and using it without taking the consent of its owner for sending out spam links, third-party threats, and other such harmful activities.

### Password Hacking



This is the process of recovering secret passwords from data that has been stored in or transmitted by a computer system.

## 1.4 TYPES OF WEB SECURITY VULNERABILITIES

### 1. SQL Injection

Injection is a security vulnerability that allows an attacker to alter backend\_statements by manipulating the user supplied data.

Injection occurs when the user input is sent to an interpreter as part of command or query and trick the interpreter into executing unintended commands and gives access to unauthorized data.

The SQL command which when executed by web application can also expose the back-end database.

### 2. Cross Site Scripting

Cross Site Scripting is also shortly known as XSS.

XSS vulnerabilities target scripts embedded in a page that are executed on the client side i.e. user browser rather than at the server side. These flaws can occur when the application takes untrusted data and send it to the web browser without proper validation.

### 3. Broken Authentication and Session Management

The websites usually create a session cookie and session ID for each valid session, and these cookies contain sensitive data like username, password, etc. When the session is ended either by logout or browser closed abruptly, these cookies should be invalidated i.e. for each session there should be a new cookie.

If the cookies are not invalidated, the sensitive data will exist in the system. For example, a user using a public computer (Cyber Cafe), the cookies of the vulnerable site sits on the system and exposed to an attacker. An attacker uses the same public computer after some time, the sensitive data is compromised.

## **4. Insecure Direct Object References**

It occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key as in URL or as a FORM parameter. The attacker can use this information to access other objects and can create a future attack to access the unauthorized data.

## **5. Cross Site Request Forgery**

Cross Site Request Forgery is a forged request came from the cross site.

CSRF attack is an attack that occurs when a malicious website, email, or program causes a user's browser to perform an unwanted action on a trusted site for which the user is currently authenticated.

A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application.

## **6. Security Misconfiguration**

Security Configuration must be defined and deployed for the application, frameworks, application server, web server, database server, and platform. If these are properly configured, an attacker can have unauthorized access to sensitive data or functionality.

Sometimes such flaws result in complete system compromise. Keeping the software up to date is also good security.

## **7. Failure to restrict URL Access**

Web applications check URL access rights before rendering protected links and buttons. Applications need to perform similar access control checks each time these pages are accessed.

In most of the applications, the privileged pages, locations and resources are not presented to the privileged users. By an intelligent guess, an attacker can access privilege pages. An attacker can access sensitive pages, invoke functions and view confidential information.

## **8. Unvalidated Redirects and Forwards**

The web application uses few methods to redirect and forward users to other pages for an intended purpose.

If there is no proper validation while redirecting to other pages, attackers can make use of this and can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

## **9. Insecure Cryptographic Storage**

Insecure Cryptographic storage is a common vulnerability which exists when the sensitive data is not stored securely.

The user credentials, profile information, health details, credit card information, etc. come under sensitive data information on a website.

This data will be stored on the application database. When this data are stored improperly by not using encryption or hashing\*, it will be vulnerable to the attackers.

(\*Hashing is transformation of the string characters into shorter strings of fixed length or a key. To decrypt the string, the algorithm used to form the key should be available)

## **10. Insufficient Transport Layer Protection**

Deals with information exchange between the user (client) and the server (application). Applications frequently transmit sensitive information like authentication details, credit card information, and session tokens over a network.

By using weak algorithms or using expired or invalid certificates or not using SSL can allow the communication to be exposed to untrusted users, which may compromise a web application and or steal sensitive information.

## **THE 6 PHASES OF ETHICAL HACKING**

### **Reconnaissance**

Reconnaissance is the process of information gathering. In this phase, the hacker gathers relevant information regarding the target system. These include detecting services, operating systems, packet-hops to reach the system, IP configuration etc. Various tools like Nmap, Hping, Google Dorks etc are used for reconnaissance purposes

### **Scanning**

In the scanning phase, the hacker begins to actively probe the target machine or network for vulnerabilities that can be exploited. Tools like Nessus, Nexpose, and NMAP are widely used by hackers in this process.

### **Gaining Access**

In this phase, the vulnerability located during scanning is exploited using various methods and the hacker tries to enter the target system without raising any alarms. The primary tool that is used in this process is Metasploit.

### **Maintaining Access**

This is one of the most integral phases. In this phase, the hacker installs various backdoors and payloads onto the target system. Just in case you don't know, Payload is a term used for

activities performed on a system after gaining unauthorized access. Backdoors help the hacker gaining quicker access onto the target system in the future.

### **Clearing Tracks**

This process is an unethical activity. It has to do with the deletion of logs of all the activities that take place during the hacking process. Nonetheless, Ethical Hackers still have to perform this phase to demonstrate how a Black Hat Hacker would go about his activities.

### **Reporting**

Reporting is the last step of finishing the ethical hacking process. Here the Ethical Hacker compiles a report with his findings and the job that was done such as the tools used, the success rate, vulnerabilities found, and the exploit processes.

## **CHAPTER 2**

### **PROJECT STATEMENT**

#### **PROBLEM STATEMENT**

To test our skills in a practical manner, we have setup a real life-like web application in the form of an online e-commerce portal. Our task is to test this e-commerce platform and find all possible vulnerabilities and loopholes in it, collect relevant Pocs and then prepare a **Detailed Developer Level Report**.

For reporting each vulnerability, we make sure that following things are mentioned

- Title of Vulnerability
- A Short Description.
- Exact URL which has the vulnerability,
- The parameters which are vulnerable (with parameter type like GET, POST Cookie, Header, etc.).
- Payload that you used to trigger the vulnerability
- Observation slides containing step by step information to replicate the exploit with Pocs.
- Business Impact of the vulnerability, explaining in detail what can be done by a hacker
- Recommendations on how to fix the vulnerability. Reputed References for the vulnerabilities

Detailed Developer Level Report with all the following vulnerability are given on following pages

- 1 S.Q.L. Injections\_
- 2 Remote File inclusion\_
- 3 Admin panel access\_\_\_\_\_



## 2.1 S.Q.L. Injections

Table 1: S.Q.L. Injections Issue

SQL injection is a code injection technique, used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution.	
S.Q.L. Injections	<p>The below mentioned URL is vulnerable to SQL injections</p> <ul style="list-style-type: none"><li>Affected URL <code>https://13.127.48.5/products.php?cat=(here)</code> <code>https://13.127.48.5/search/search.php?q=(here)</code></li><li>Affected parameters 1.cat 2.q</li><li>Payload <code>cat=2'</code> <code>q=adidas'</code></li></ul>

### Observations

At home page click on any 1 category.

Notice the get category of cat and add ' and then observe the error.



Fig 1 : observations

### Proof of concept

Attacker can execute the sql commands as shown below and access confidential data.

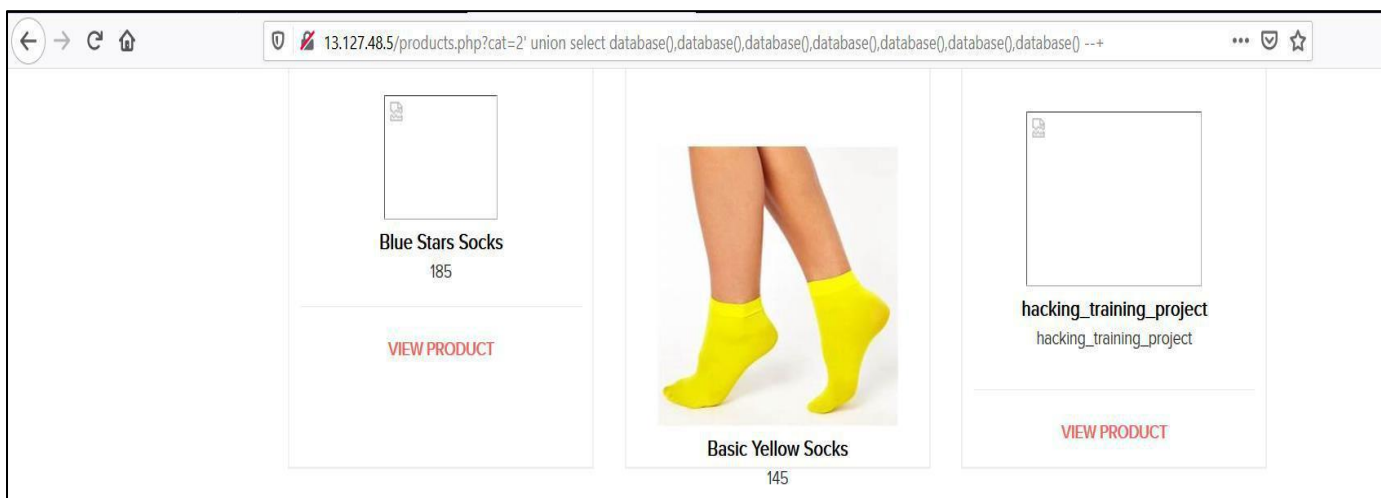
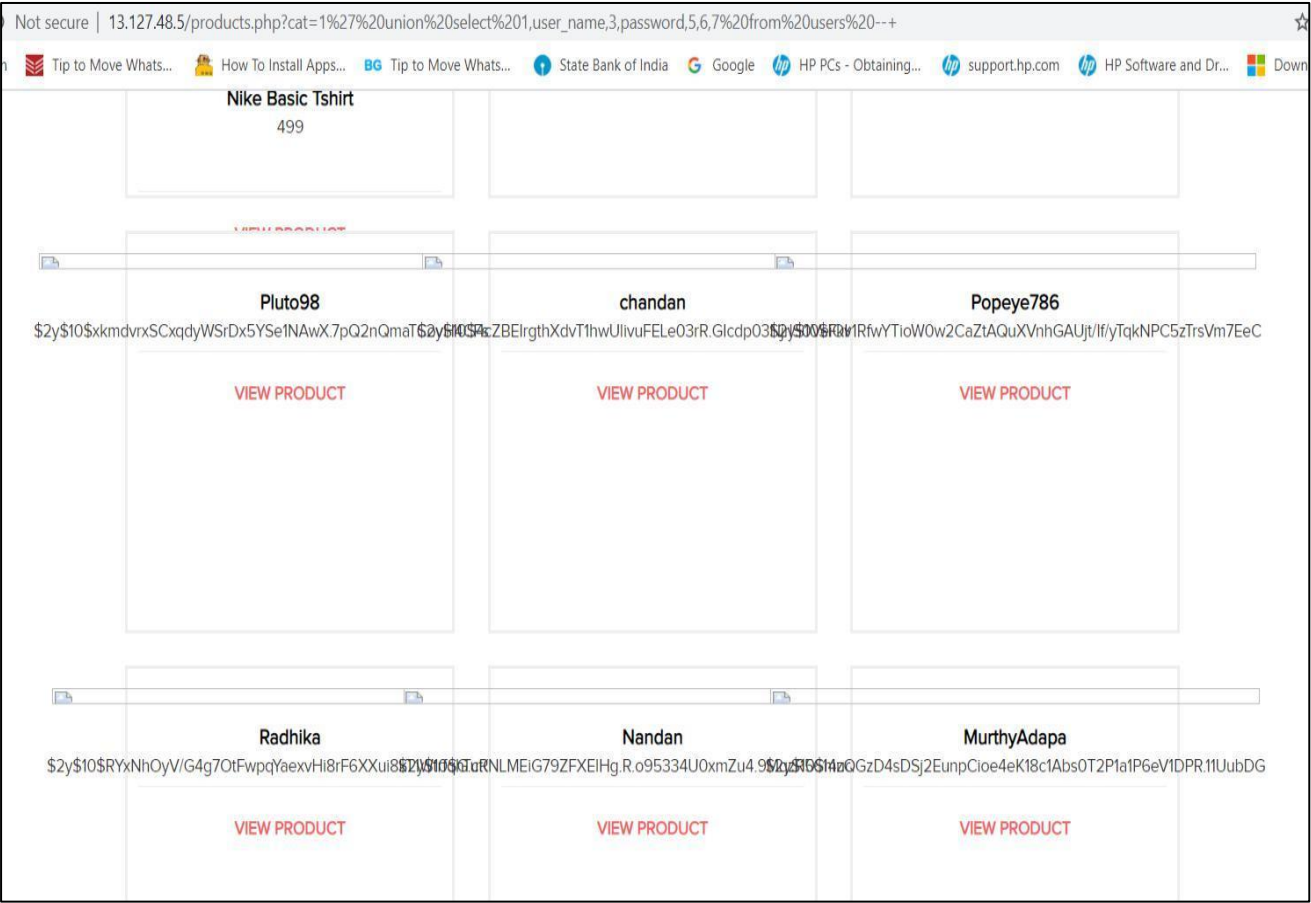


Fig 2: Proof of concept

**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.



**Fig 3: Business impact**

## 2.2 Remote File inclusion

Table 2: Remote File inclusion Issue

Remote file inclusion (RFI) is an attack targeting vulnerabilities in web applications that dynamically reference external scripts.	
RFI	<div>Below mentioned url is vulnerable to RFI</div> <ul style="list-style-type: none"><li>Affected url 52.66.206.249/?includelang=(here)</li><li>Payload ../../../../../../etc/passwd https://google.co.in</li></ul>

### Observations

When you click on change language you get a 'get' parameter of includelang which is vulnerable for file inclusion.

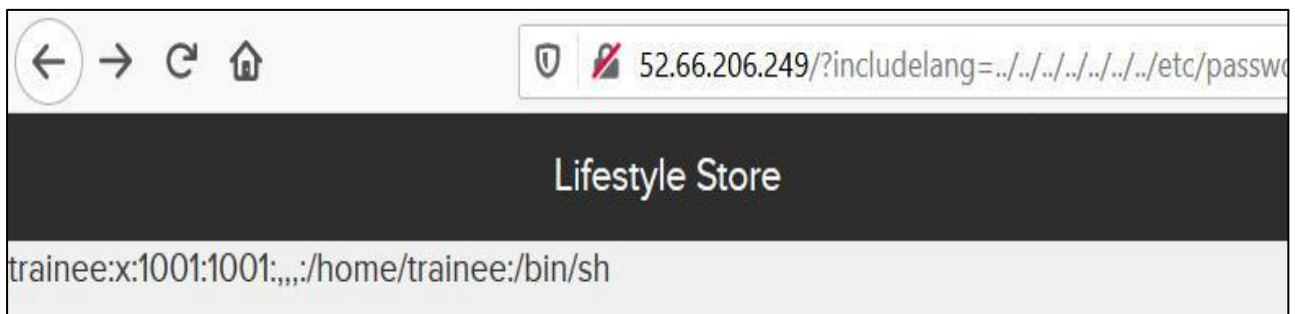
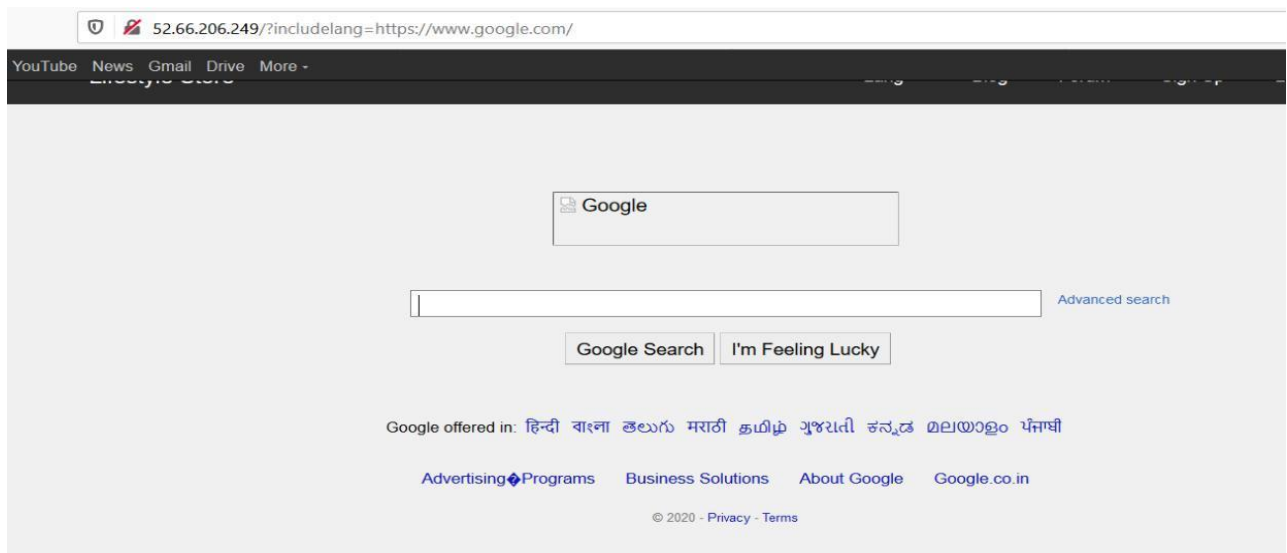


Fig 4 : observations

Here we have tested Local file inclusion to execute a file which gives us user name

### POC-attacker can upload shells

**Attacker can exploit** the referencing function in an application to upload malware (e.g., backdoor shells) from a remote URL located within a different domain.



**Fig 5: Proof of concept**

### **Business impact- Extremely high**

Any attacker can have the root access of of your website

He can execute commands

Through the website he can have access of the server and can infect other websites hosted on that server

He can even deface your website

## 2.3 Admin panel access

Table 3: Admin panel access Issue

the login panel for administration of the sites is left publicly and easily accessible, either through easy to guess URLs or unpatched vulnerabilities.	
Admin panel access	<p>admin panel of this website can easily be taken over by brute forcing O.T.P.</p> <ul style="list-style-type: none"><li>Affected url <a href="http://13.126.208.41/reset_password/admin.php">http://13.126.208.41/reset_password/admin.php</a></li><li>Payload 001-999 digits</li></ul>

### Observation

In the admin login section there is a reset admin option which only needs a 3-digit otp

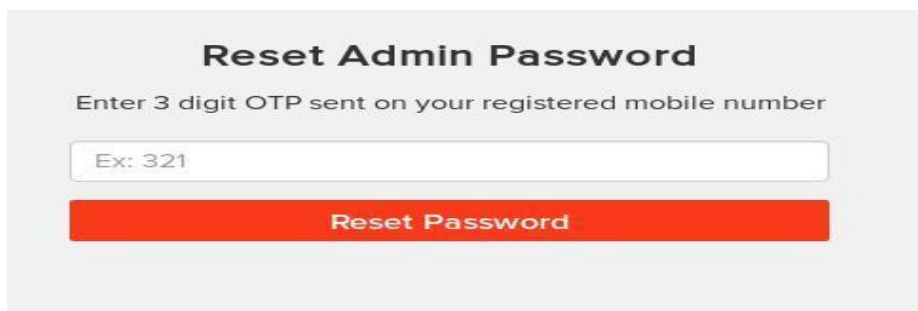


Fig 6 : observations

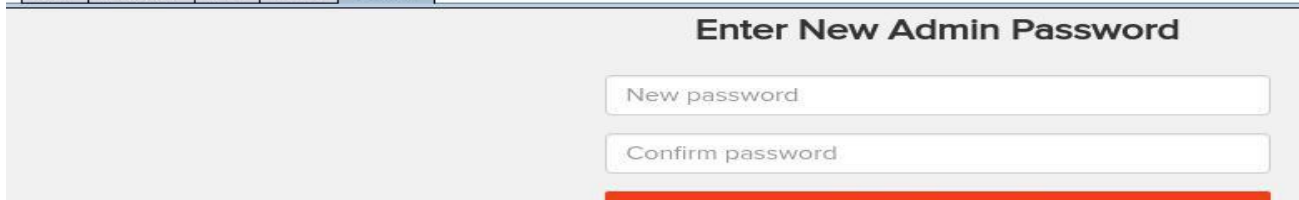
### POC

looking for the otp correct value:-

Request	Payload	Status	Error	Timeout	Length	Comment
153	152	200	<input type="checkbox"/>	<input type="checkbox"/>	4476	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
1	000	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
4	003	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
3	002	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
6	005	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
8	007	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
9	008	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
11	010	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
10	009	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
12	011	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
14	013	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
13	012	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
7	006	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
17	016	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
18	017	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
16	015	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	

Request	Response
Raw	Headers
Hex	HTML
Render	

Here is the admin dashboard

13.126.208.41/admin31/dashboard.php

Lifestyle Store Dashboard Logout

### Admin Dashboard

CONSOLE

Add Product:

No.	Product Name	Product Description	Seller	Category	Image	Price	
			<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input checked="" type="radio"/> T Shirt <input type="radio"/> Socks <input type="radio"/> Shoes	<input type="text" value="UPLOAD"/>	<input type="text"/>	<input type="button" value="Add"/>

Fig 7: Proof of concept

## Business impact-extremely high

He can change the rates of items selling on the web sites

- He can add and delete the items
- He can change the seller and catagories
- He can execute commands on the server through console options, which can be further used to harm your website

## Recommendations

- The first is to implement an account lockout policy. For example, after three failed login attempts, the account is locked out until an administrator unlocks it.
- Tools such as the free [reCAPTCHA](#) can be used to require the user to enter a word or solve a simple math problem to ensure the user is, in fact, a person.
- Admin login page should be hidden very securely,
- The otp should be alpha numeric and at least of 6-letters and digits.

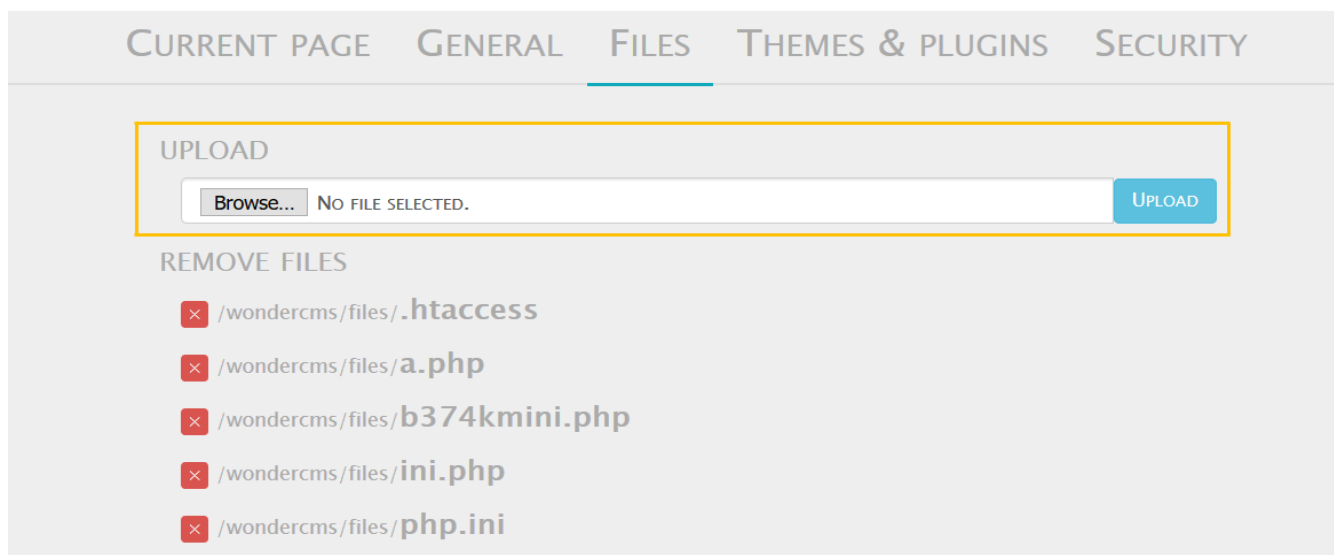
## 2.4 Insecure file uploads

Table 4: Insecure file uploads Issue

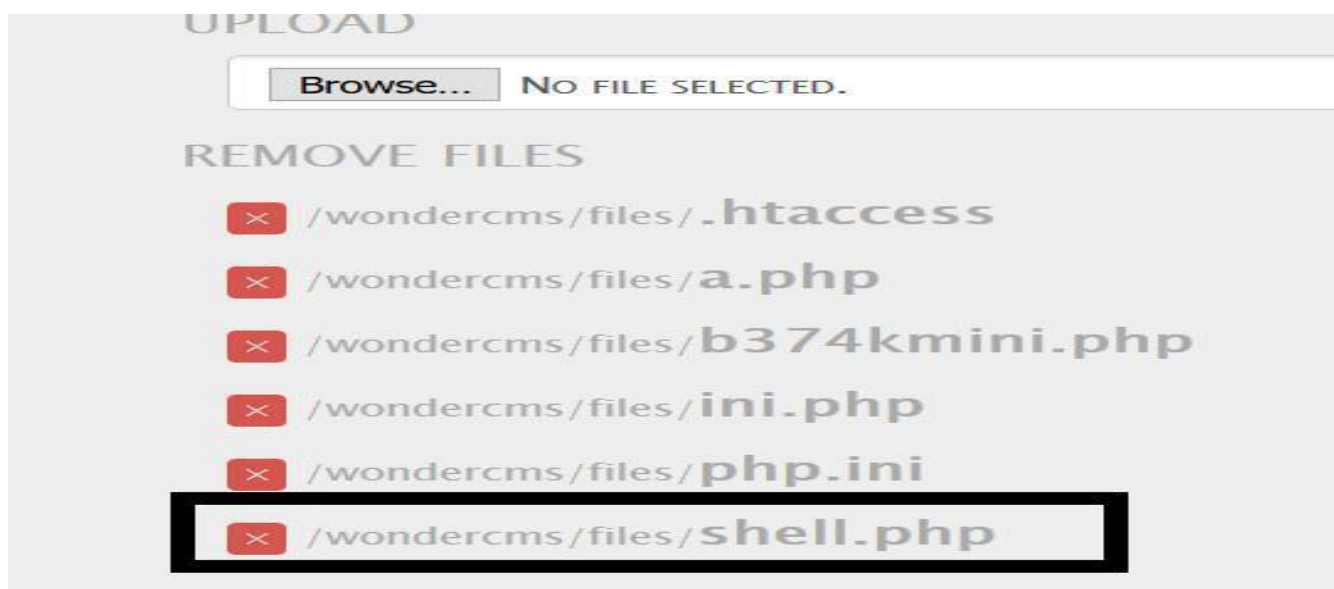
A remote file upload vulnerability is a vulnerability where an application uses user input to fetch a remote file from a site on the Internet and store it locally.	
Insecure file uploads	<p>The url given below is vulnerable to insecure file uploads</p> <ul style="list-style-type: none"><li>Affected url <code>http://13.233.83.32/wondercms/</code></li><li>Uploaded file backdoor shell</li></ul>

### Observations

- In the blog page of website there is a upload option in the settings



- I tried uploading a shell and I was successful



- The shell I uploaded was executed successfully



**Fig 8 : observations**

## **Business impact-Extremely high**

The consequences of unrestricted file upload can vary: -

- including complete system takeover, an overloaded file system or database.
- forwarding attacks to back-end systems
- client-side attacks, or simple defacement.

It depends on what the application does with the uploaded file and especially where it is stored.

## **Recommendations**

- ◆ The file types allowed to be uploaded should be restricted to only those that are necessary for business functionality.
- ◆ Never accept a filename and its extension directly without having a whitelist filter.
- ◆ All the control characters and Unicode and the special characters should be discarded



## 2.5 Seller account access

Table 5 : Seller account access Issue

Access control attacks typically circumvent or bypass access control methods to steal data or user credentials.	
Seller account access	<p>the default page given below shows the seller accounts and passwords</p> <p>Seller account access</p> <ul style="list-style-type: none"><li>Affected url</li></ul> <p><a href="http://13.233.83.32/userlist.txt">http://13.233.83.32/userlist.txt</a></p>

### Observations

At the homepage after adding userlist.txt the following page is opened

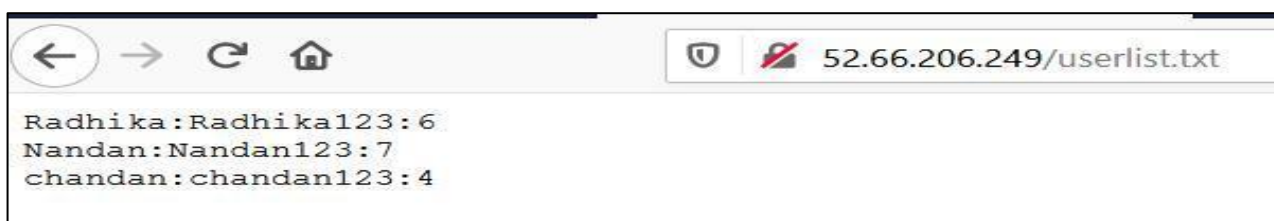


Fig 9 : observations

### Proof of concept

POC-attacker has the seller dashboard access

On entering the credentials in the seller account login we have accessed the dashboard



Fig 10 : Proof of concept

### Business impact-Extremely high

Attacker can access the seller dashboard and then can edit the items he is selling

### Recommendations

The developer should disable these confidential default pages

## 2.6 Default admin password

Table 6 : Default admin password Issue

A Default Credential vulnerability is a type of vulnerability that is most commonly found to affect the devices like modems, routers, digital cameras, and other devices having some pre-set (default) administrative credentials to access all configuration settings.	
Default admin password	<p>The url given below is using the default admin credentials</p> <ul style="list-style-type: none"><li>Affected url <a href="http://52.66.65.223/ovidentiaCMS/index.php?tg=login&amp;cmd=authform&amp;msg=Connexion&amp;err=&amp;restricted=1">http://52.66.65.223/ovidentiaCMS/index.php?tg=login&amp;cmd=authform&amp;msg=Connexion&amp;err=&amp;restricted=1</a></li><li>Component name ovidentia content management system</li></ul>

### Observations

In the ovidentia cms page, there is option to login as admin .On clicking it we saw this page.

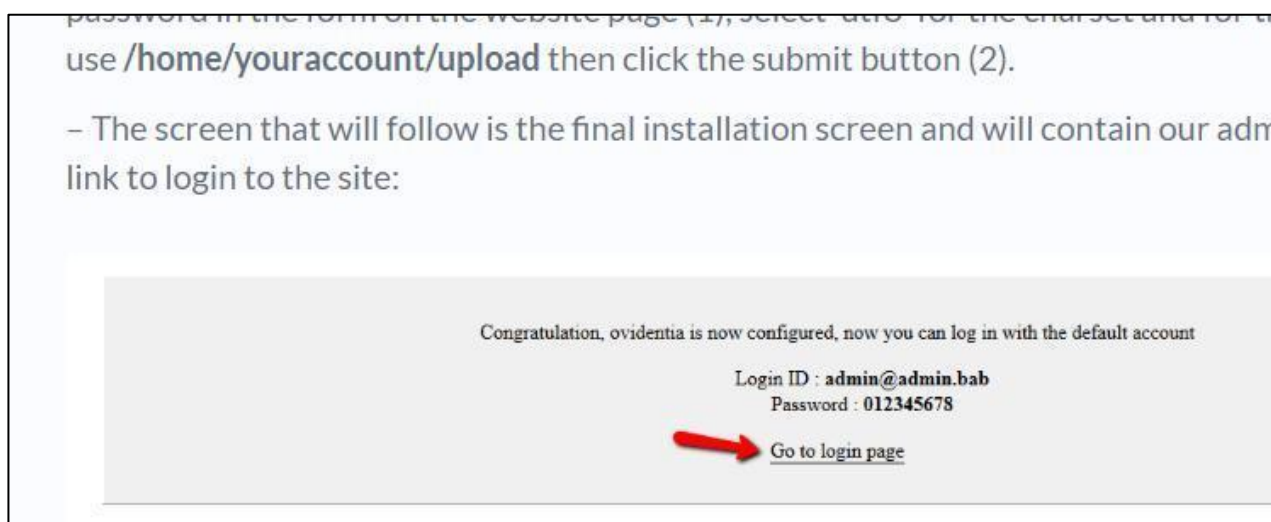


Fig 11 : observations

### Proof of concept

POC-ovidentia admin access

In searching for default ovidentia admin credentials we get



We got the admin access



### **Business impact- Extremely high**

- Attacker will have all the admin privileges
- He can easily deface the ovidentia CMS

### **Recommendations**

- Disable the default debug pages
- Hide the admin login page
- Disable the default passwords and use a strong username and password

## 2.7 Components with known vulnerability

Table 7 Components with known vulnerability Issue

components such as libraries and frameworks used within the app almost always execute with full privileges. If a vulnerable component is exploited, it makes the hacker's job easier to cause a serious data loss or server takeover.	
Components with known vulnerability	<p>The urls given below are of the components with known vulnerability</p> <ul style="list-style-type: none"><li>Affected url <a href="http://52.66.65.223/wondercms/">http://52.66.65.223/wondercms/</a> <a href="http://52.66.65.223/forum/">http://52.66.65.223/forum/</a> And PHP</li></ul>

### Observations

- I checked the versions of these components they were out dated



- It was 2015 version of codoform was 3.0

Key Facts	
CMS name	WonderCMS
Current version (stable)	2.5.1
Latest release date (stable)	05/03/2018

Codoform v.4.6 released - A  
<https://codologic.com> › forum › topic

- The php version of this website is 5.6.39-1 which is out dated

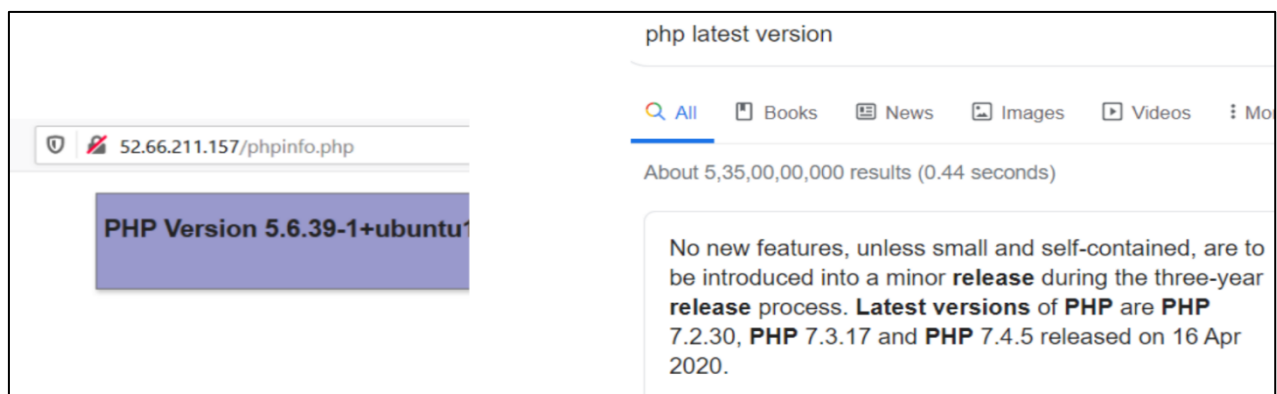


Fig 13 : observations

## Proof of concept

Both the components have known public exploits

Codoforum : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentic
1	<a href="#">CVE-2014-9261</a>	<a href="#">22</a>	1	Dir. Trav.	2015-03-23	2015-03-24	5.0	None	Remote	Low	Not req

The sanitize function in Codoforum 2.5.1 does not properly implement filtering for directory traversal sequences, which allows remote attackers to read arbitrary files via a .. (dot dot)

Wondercms : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-2019-5956</a>	<a href="#">22</a>		Dir. Trav.	2019-09-12	2019-09-13	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Directory traversal vulnerability in WonderCMS 2.6.0 and earlier allows remote attackers to delete arbitrary files via unspecified vectors.														
2	<a href="#">CVE-2018-1000062</a>	<a href="#">79</a>		XSS	2018-02-09	2018-03-05	3.5	None	Remote	Medium	Single system	None	Partial	None
WonderCMS version 2.4.0 contains a Stored Cross-Site Scripting on File Upload through SVG vulnerability in uploadFileAction(). 'svg' => 'image/svg+xml' that can result in An attacker can execute arbitrary script on an unsuspecting user's browser. This attack appear to be exploitable via Crafted SVG File.														
3	<a href="#">CVE-2018-14387</a>	<a href="#">384</a>			2018-07-18	2018-09-19	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
An issue was discovered in WonderCMS before 2.5.2. An attacker can create a new session on a web application and record the associated session identifier. The attacker then causes the victim to authenticate against the server using the same session identifier. The attacker can access the user's account through the active session. The Session fixation attack fixes a session on the victim's browser, so the attack starts before the user logs in.														
4	<a href="#">CVE-2018-7172</a>	<a href="#">22</a>		Dir. Trav.	2018-02-27	2018-03-23	5.5	None	Remote	Low	Single system	None	Partial	Partial
In index.php in WonderCMS before 2.4.1, remote attackers can delete arbitrary files via directory traversal.														
5	<a href="#">CVE-2017-14527</a>	<a href="#">74</a>			2018-01-26	2019-04-30	5.0	None	Remote	Low	Not required	None	Partial	None
** DISPUTED ** WonderCMS 2.3.1 is vulnerable to an HTTP Host header injection attack. It uses user-entered values to redirect pages. NOTE: the vendor reports that exploitation is unlikely because the attack can only come from a local machine or from the administrator as a self attack.														
6	<a href="#">CVE-2017-14522</a>	<a href="#">79</a>		XSS	2018-01-26	2018-02-14	4.3	None	Remote	Medium	Not required	None	Partial	None
** DISPUTED ** In WonderCMS 2.3.1, the application's input fields accept arbitrary user input resulting in execution of malicious JavaScript. NOTE: the vendor disputes this issue stating that this is a feature that enables only a logged in administrator to write execute JavaScript anywhere on their website.														
7	<a href="#">CVE-2017-14521</a>	<a href="#">434</a>			2018-01-26	2019-04-26	6.5	None	Remote	Low	Single system	Partial	Partial	Partial
In WonderCMS 2.3.1, the upload functionality accepts random application extensions and leads to malicious File Upload.														
8	<a href="#">CVE-2017-7951</a>	<a href="#">352</a>		CSRF	2017-04-20	2017-04-24	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
WonderCMS before 2.0.3 has CSRF because of lack of a token in an unspecified context.														
9	<a href="#">CVE-2014-8705</a>	<a href="#">20</a>		Exec Code File Inclusion	2017-03-17	2017-03-20	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
PHP remote file inclusion vulnerability in editInplace.php in Wonder CMS 2014 allows remote attackers to execute arbitrary PHP code via a URL in the hook parameter.														
10	<a href="#">CVE-2016-8704</a>	<a href="#">22</a>		Dir. Trav.	2017-03-17	2017-03-20	7.5	None	Remote	Low	Not required	Partial	Partial	Partial

Severity

Patch available

Number of vulnerabilities

CVE ID

CWE ID

High

YES

20

CVE-2018-19935  
CVE-2019-6977  
CVE-2016-10166

CWE-476  
CWE-125  
CWE-122  
CWE-617  
CWE-120  
CWE-388  
CWE-787  
CWE-191  
CWE-264  
CWE-835

## **Business impact- Extremely high**

Anyone can perform any attacks (available) as all the exploits are available publicly.

- It can cause severe damage to the website
- He may be able to upload backdoor shells
- He will easily deface your website

## **Recommendations**

- Update all the components and the php version which is running on it
- Hide the current versions info from there pages

## 2.8 Customer account access

**Table 8: Customer account access Issue**

Access control vulnerabilities occur when users can act outside of their intended permissions. This typically leads to unauthorized access, information disclosure, and modification or destruction of data. These vulnerabilities arise from insecure coding or insecure implementation of authentication and authorization mechanisms.

Customer account access	<p>The url given bellow contains the is giving a descriptive error which change password option</p> <p style="text-align: center;"><b>CUSTOMER ACCOUNT ACCESS</b></p> <ul style="list-style-type: none"> <li>Affected url <a href="http://52.66.65.223/reset_password/customer.php?username=Donal234">http://52.66.65.223/reset_password/customer.php?username=Donal234</a></li> <li>User names    Dnal234                     Pluto98                     Popeye786</li> </ul>
-------------------------	---

### Observations

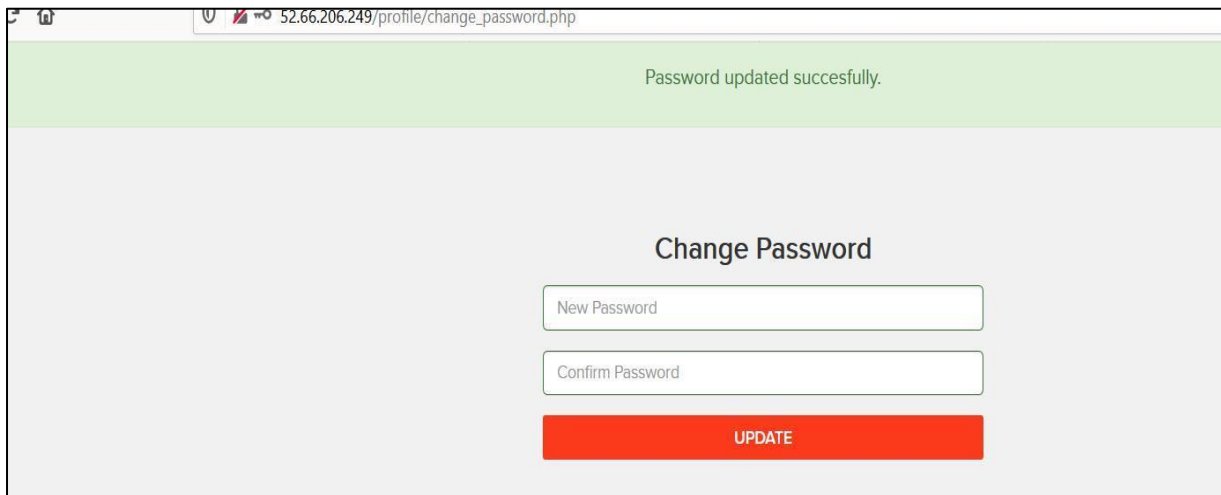
- In the forgot password option only username is required to change password

- On entering the username it gives the change password link on email which can be edited by burp suite

```
ring(20) "hackinglab1@zoho.com" object(PHPMailer\PHPMailer\Exception)#6 (7) { ["message":protected]=> string(30) "SMTP Error: data not accepted." ["string":"Exception":private]=> string(0) "" ["code":protected]=> int(2) ["file":protected]=> string(69) "/var/www/hacking_project/vendor/phpmailer/phpmailer/src/PHPMailer.php" ["line":protected]=> int(1806) ["trace":"Exception":private]=> array(3) { [0]=> array(6) { ["file"]=> string(69) "/var/www/hacking_project/vendor/phpmailer/phpmailer/src/PHPMailer.php" ["line"]=> int(1516) ["function"]=> string(8) "smtpSend" ["class"]=> string(29) "PHPMailer\PHPMailer\PHPMailer" ["type"]=> string(2) ">" ["args"]=> array(2) { } } } ["Date":protected]=> string(48) "Date: Sat, 4 Jul 2020 00:22:42 +0530 To: donald@lifestylestore.com From: Hackinglab Reply-To: No Reply Subject: Password reset request Message-ID: X-Mailer: PHPMailer 6.0.6 (https://github.com/PHPMailer/PHPMailer) MIME-Version: 1.0 Content-Type: multipart/alternative; boundary="b1_BbGOk0Ky81uFTaVe1bzbbgftsMI2KJKu5u2I3eymD5w" Content-Transfer-Encoding: 8bit " [1]> string(582) "This is a multi-part message in MIME format. --b1_BbGOk0Ky81uFTaVe1bzbbgftsMI2KJKu5u2I3eymD5w Content-Type: text/plain; charset=us-ascii Copy and paste this url http://52.66.206.249/reset_password/verify.php?key=778522555c6669996f5a24.34991684 in your browser's address bar to reset your password --b1_BbGOk0Ky81uFTaVe1bzbbgftsMI2KJKu5u2I3eymD5w Content-Type: text/html; charset=us-ascii Click here to reset your password b1_BbGOk0Ky81uFTaVe1bzbbgftsMI2KJKu5u2I3eymD5w--" } } [1]> array(6) { ["file"]=> string(69) "/var/www/hacking_project/vendor/phpmailer/phpmailer/src/PHPMailer.php" ["line"]=> int(1352) ["function"]=> string(8) "postSend" ["class"]=> string(29) "PHPMailer\PHPMailer\PHPMailer" ["type"]=> string(2) ">" ["args"]=> array(0) { } } [2]> array(6) { ["file"]=> string(52) "/var/www/hacking_project/reset_password/customer.php" ["line"]=> int(51) ["function"]=> string(4) "send" ["class"]=> string(29) "PHPMailer\PHPMailer\PHPMailer" ["type"]=> string(2) ">" ["args"]=> array(0) { } } ["previous":"Exception":private]> NULL }
```



- After entering another email address password can be changed easily



The screenshot shows a web browser window with the address bar displaying '52.66.206.249/profile/change\_password.php'. The page has a light green header bar with the text 'Password updated succesfully.' Below this, the main content area is light gray and contains a 'Change Password' form. The form has two input fields: 'New Password' and 'Confirm Password', both with green borders. Below these fields is a red button with the text 'UPDATE' in white capital letters.

**Fig 15 : observations**

### **Business impact –extremely high**

- Attacker can get the customer account access
- Then he can make changes on it like changing the personnel details, cancel the orders , etc
- This will reduce your organizations reputation

### **Recommendations**

- You should include the otp option and make it compulsory
- Security checks on the server side should be done completely
- Captcha                      option                      should                      also                      be                      included



## 2.9 Forced browsing

Table 9: Forced browsing Issue

Forced browsing is an attack where the aim is to enumerate and access resources that are not referenced by the application, but are still accessible.	
Forced browsing	The below mentioned URL is vulnerable to forced browsing Affected url- <a href="http://52.66.65.223/">http://52.66.65.223/</a> Forced url- <a href="http://52.66.65.223/admin31/dashboard.php">http://52.66.65.223/admin31/dashboard.php</a>

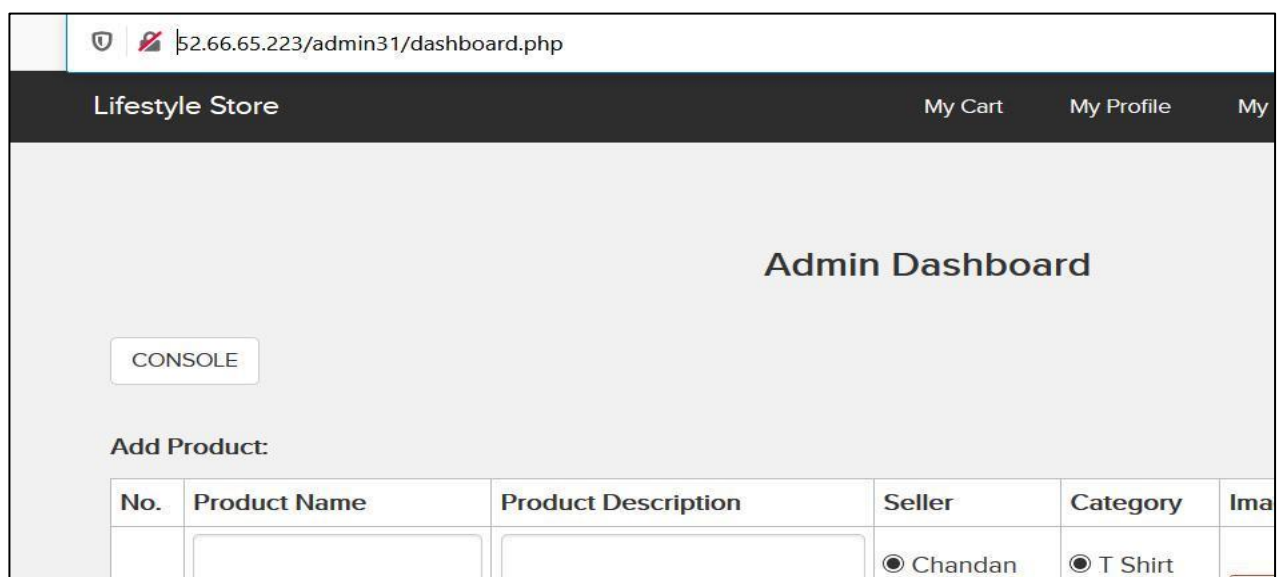
### observations

When I tried to go in admin dashboard without logging in I was successful

### Proof of concept

POC-admin dashboard access

Here is the admin dashboard just by entering its complete url



- He can edit all the items
- He can execute any harmful command through console

### Recommendations

- Server side security checks should be performed perfectly
- Make the admin page url complicated so that it couldn't be guessed

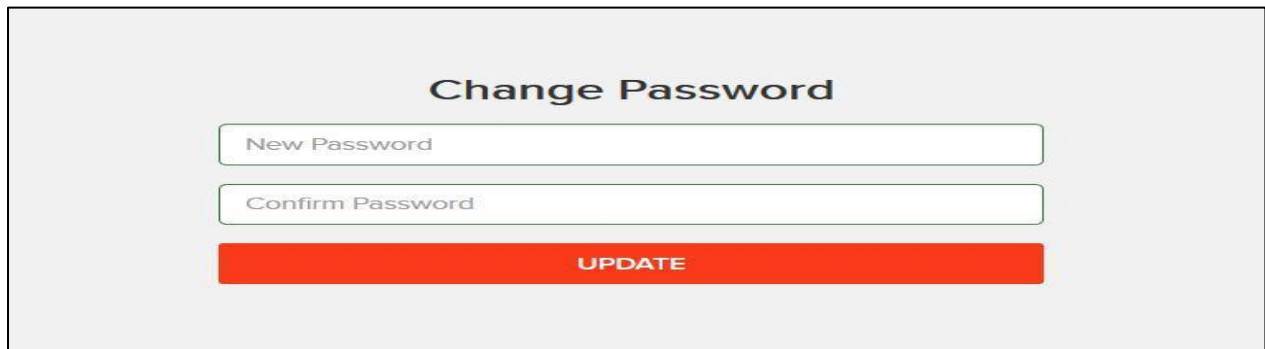
## 2.10 C.S.R.F.

Table 10 : C.S.R.F. issue

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.	
C.S.R.F	The url given below is vulnerable to CSRF  Affected url <a href="http://52.66.65.223/profile/change_password.php">http://52.66.65.223/profile/change_password.php</a> <a href="http://52.66.65.223/cart/cart.php">http://52.66.65.223/cart/cart.php</a>

### Observations-1

- There is a change password option in profile page



### Proof of concept-1

- Make a html page to change username and password

```
<html>
<head>
<title> CSRF POC </title>
</head>
<body>
  <form name='change-password' id='change-password' method='POST' action='http://52.66.65.223/profile/change_password_submit.php'>
    <input type='password' placeholder="New Password" name="password" id="password" value="1234">
    <input type='password' placeholder="Confirm Password" name="password_confirm" id="password_confirm" value="1234">
    <button type='submit' class="btn btn-primary">Update</button>
  </form>
</body>
</html>
```

On clicking the update button we get success



## Observations-2

- There is a confirm button in my orders

S.No	Product	Price
1	Adidas Socks - Pack <a href="#">Remove</a>	450
	Total	450

**Have a coupon?**

Apply

Your coupon should look like UL\_6666

---

**Shipping Details**  
Brutus  
A-56 Sailor's ship, popeyeworld

**Payment Mode**  
☒ Cash on delivery

**CONFIRM ORDER**

### Proof of concept-2

- Make a html page to confirm order

```
ical hacking\LifeStyle_Store\Vulnerabilities\CSRF\cart.html
<head>
  <title> CSRF POC </title>
</head>
<body>
  <form action="http://52.66.65.223/orders/confirm.php" method='POST'>
    <input type='Submit' value="Submit Request"></input>
  </body>
</html>
```

**Order Id: BD21907B81EA**

---

**PRODUCTS:**

Adidas Socks - Pack	INR 450
<b>Total</b>	<b>INR 450</b>

**SHIPPING DETAILS:**  
Name - Brutus  
Email - Pluto@lifestylestore.com  
Phone - 8912345670  
Address - A-56 Sailor's ship, popeyeworld

**PAYMENT MODE**  
Cash on delivery

---

Order placed on : 2020-07-07 20:55:51Status: DELIVERED

### Business impact- severe

- Attacker can change the password by uploading phishing pages
- Attacker can confirm the order without consent of user

### Recommendations

- Use of tokens and session cookies
- Referrer header should be checked at server side

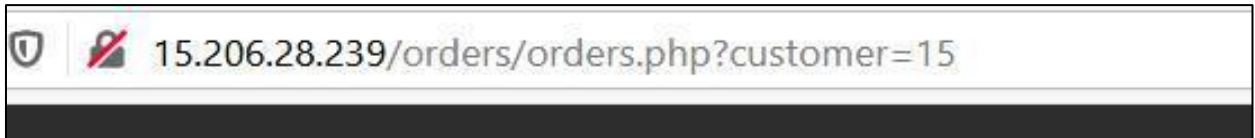
## 2.11 Insecure direct object references (IDOR)

Table 11 : Insecure direct object references (IDOR) issue

IDOR (Insecure Direct Object Reference) is a common vulnerability that occurs when a reference to an internal implementation object is exposed without any other access control. The vulnerability is often easy to discover and allows attackers to access unauthorized data.	
Insecure direct object references (IDOR).	<p>The below mentioned URL is vulnerable to IDOR</p> <p>Affected URL  <a href="http://15.206.28.239/orders/orders.php?customer=(here)">http://15.206.28.239/orders/orders.php?customer=(here)</a></p> <p>Payload 0-50</p>

## Observations

In my orders page I saw customer number in URL



Brute forcing it

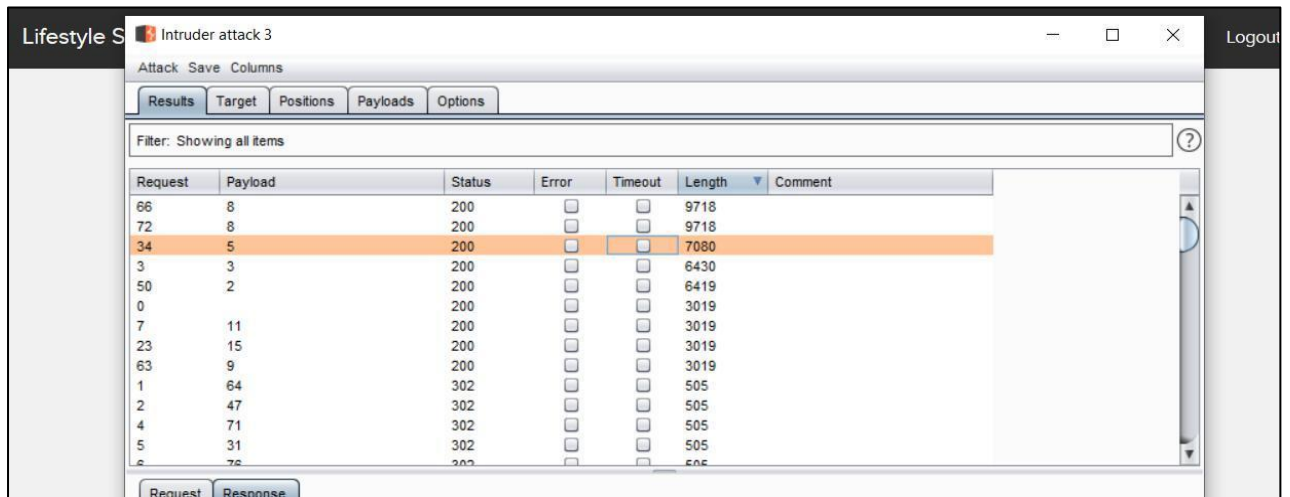


Fig 21 : observations

## Proof of concept

- I got other customers and their order details

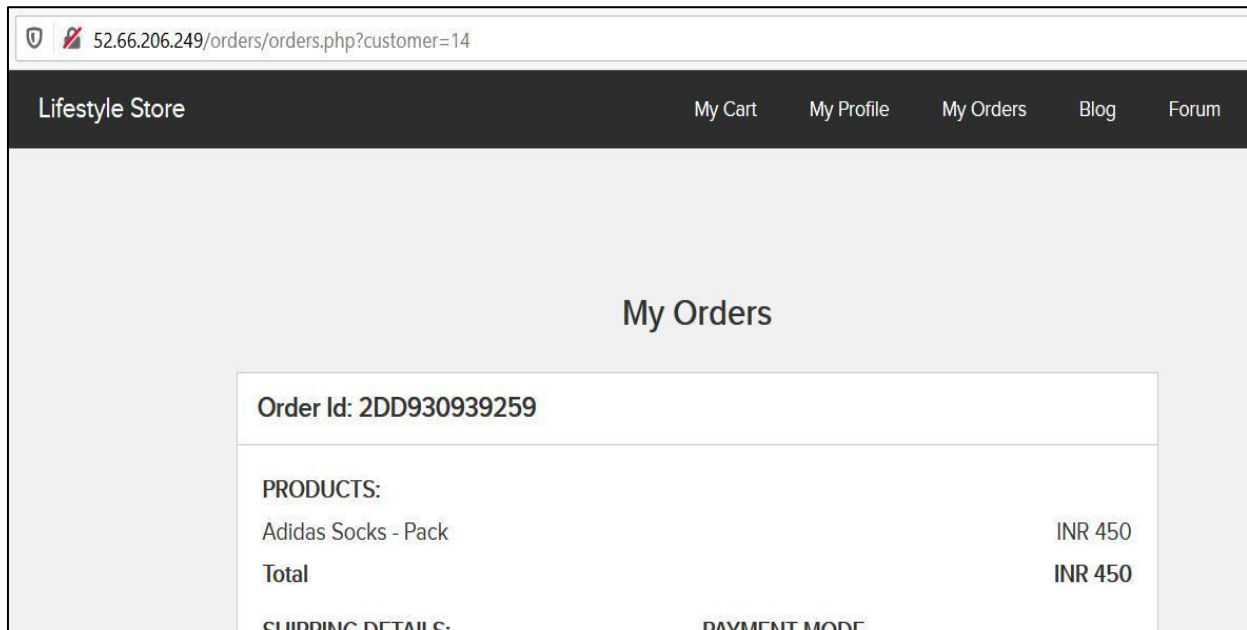


Fig 22 : Proof of concept

## Recommendations

- Instead of requiring the references in the URL, use the information already present in the user's session on the server to locate the resources to serve.
- If it is not possible to avoid exposing the references to objects in the URL, as explained earlier, the *indirect reference map* technique is helpful. The idea behind it is to substitute the sensitive direct internal reference in URL parameters or form fields with a random value that is difficult to predict (such as a GUID) or specific only to the logged-in user

## 2.11 Open redirection

Table 12 : Open redirection issue

Open redirection vulnerabilities arise when an application incorporates user-controllable data into the target of a redirection in an unsafe way. An attacker can construct a URL within the application that causes a redirection to an arbitrary external domain.	
Open redirection	<ul style="list-style-type: none"><li>The URL given below Is vulnerable to open redirection</li><li>Affected URL <code>http://13.127.179.208/redirect.php?url=(www.radhikafancystore.com)</code></li></ul>

### Observations

- On clicking the brand website redirection occurs



Fig 23 : observations

On changing the link to google.co.in we were redirected to it



### Business impact- severe

- He can access the users personnel credentials which would be very harmful
- They can redirect your page to a malware site
- They can redirect you to phishing pages

### Recommendations

- Design your app to avoid URL redirects or forwards as a best practice. If unavoidable, encrypt the target URL such that the URL: token mapping is validated on the server.
- Verify URL patterns using regular expressions to check if they belong to valid URLs. However, malicious URLs can pass that check.

## CHAPTER 3

### RESULT N ASSESSMENT

#### 3.1 Vulnerabilities index

SL. No.	Severity	Vulnerabilities	count
1	Critical	SQL injections	2
2	Critical	Remote file inclusion	1
3	Critical	Admin panel access	1
4	Critical	Insecure file uploads	1
5	Critical	Seller account access	1
6	Critical	Default admin password	1
7	Critical	Components with known	3
8	Critical	Customer account access	1
9	Severe	Forced browsing	1
10	Severe	C.S.R.F	2
11	Severe	Insecure direct object ref.	1
12	Severe	Open redirection	1



### **3.2 Project Solution**

The project web application that I was assigned to, had 16 vulnerabilities.

Here is the breakdown of the various types of vulnerabilities that were present in the web application:

- 1 SQL injection - 2
- 2 Remote file inclusion - 1
- 3 Insecure Direct Object Reference – 1
- 4 Admin panel access-1
- 5 Insecure file uploads-1
- 6 Seller account access-1
- 7 Default admin password-1
- 8 Components with known vulnerability-3
- 9 Customer account access-1
- 10 Forced browsing-1
- 11 C.S.R.F-2
- 12 Insecure direct object ref.-1
- 13 Open redirection-1

## **CHAPTER-4**

### **CONCLUSION AND RECOMMENDATION**

After reviewing the vulnerability and getting access of the shopping website is a big security threat. website data can be misused. So it should be patched and updated with the latest software.

#### **Recommendations to improve web security**

- ❖ Where possible, implement multi-factor authentication to prevent automated, credential stuffing, brute force, and stolen credential re-use attacks.
- ❖ Do not ship or deploy with any default credentials, particularly for admin users.
- ❖ Implement weak-password checks, such as testing new or changed passwords against a list of the Align password length, complexity and rotation policies with \_ or other modern, evidence based password policies.
- ❖ Ensure registration, credential recovery, and API pathways are hardened against account enumeration attacks by using the same messages for all outcomes.
- ❖ Limit or increasingly delay failed login attempts. Log all failures and alert administrators when credential stuffing, brute force, or other attacks are detected.
- ❖ Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login. Session IDs should not be in the URL, be securely stored and invalidated after logout, idle, and absolute timeouts.

*Do the following, at a minimum, and consult the references:*

- ❖ Classify data processed, stored or transmitted by an application. Identify which data is sensitive according to privacy laws, regulatory requirements, or business needs.
- ❖ Apply controls as per the classification.
- ❖ Don't store sensitive data unnecessarily. Discard it as soon as possible or use PCI DSS compliant tokenization or even truncation. Data that is not retained cannot be stolen.
- ❖ Make sure to encrypt all sensitive data at rest.
- ❖ Unique application business limit requirements should be enforced by domain models.
- ❖ Disable web server directory listing and ensure file metadata (e.g.. git) and backup files are not present within web roots.
- ❖ Log access control failures, alert admins when appropriate (e.g. repeated failures).
- ❖ Rate limit API and controller access to minimize the harm from automated attack tooling.

## **REFERENCES**

- ❖ [https://owasp.org/www-project-top-ten/2017/A1\\_2017-Injection](https://owasp.org/www-project-top-ten/2017/A1_2017-Injection)
- ❖ [https://owasp.org/www-project-top-ten/2017/A2\\_2017-Broken\\_Authentication](https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication)
- ❖ [https://owasp.org/www-project-top-ten/2017/A3\\_2017-Sensitive\\_Data\\_Exposure](https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure)
- ❖ [https://owasp.org/www-project-top-ten/2017/A5\\_2017-Broken\\_Access\\_Control](https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control)
- ❖ [https://owasp.org/www-project-top-ten/2017/A6\\_2017-Security\\_Misconfiguration](https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration)
- ❖ [https://owasp.org/www-project-top-ten/2017/A8\\_2017-Insecure\\_Deserialization](https://owasp.org/www-project-top-ten/2017/A8_2017-Insecure_Deserialization)
- ❖ [https://owasp.org/www-project-top-ten/2017/A9\\_2017-Using\\_Components\\_with\\_Known\\_Vulnerabilities](https://owasp.org/www-project-top-ten/2017/A9_2017-Using_Components_with_Known_Vulnerabilities)
- ❖ [https://owasp.org/www-project-top-ten/2017/A10\\_2017-Insufficient\\_Logging%2526Monitoring](https://owasp.org/www-project-top-ten/2017/A10_2017-Insufficient_Logging%2526Monitoring)
- ❖ <https://www.greycampus.com/opencampus/ethical-hacking/web-server-and-its-types-of-attacks>
- ❖ <https://trainings.internshala.com/>