Subject: **Advisory-Prevention against ISSB Website Exploitation. (Advisory No. 5)**

1. During Cyber scanning, ISSB website **(issb.com.pk)** has been found vulnerable to exploitation. Following critical vulnerabilities have been identified: -

   a. Blind SQL Injection

   b. Directory Listing Enabled

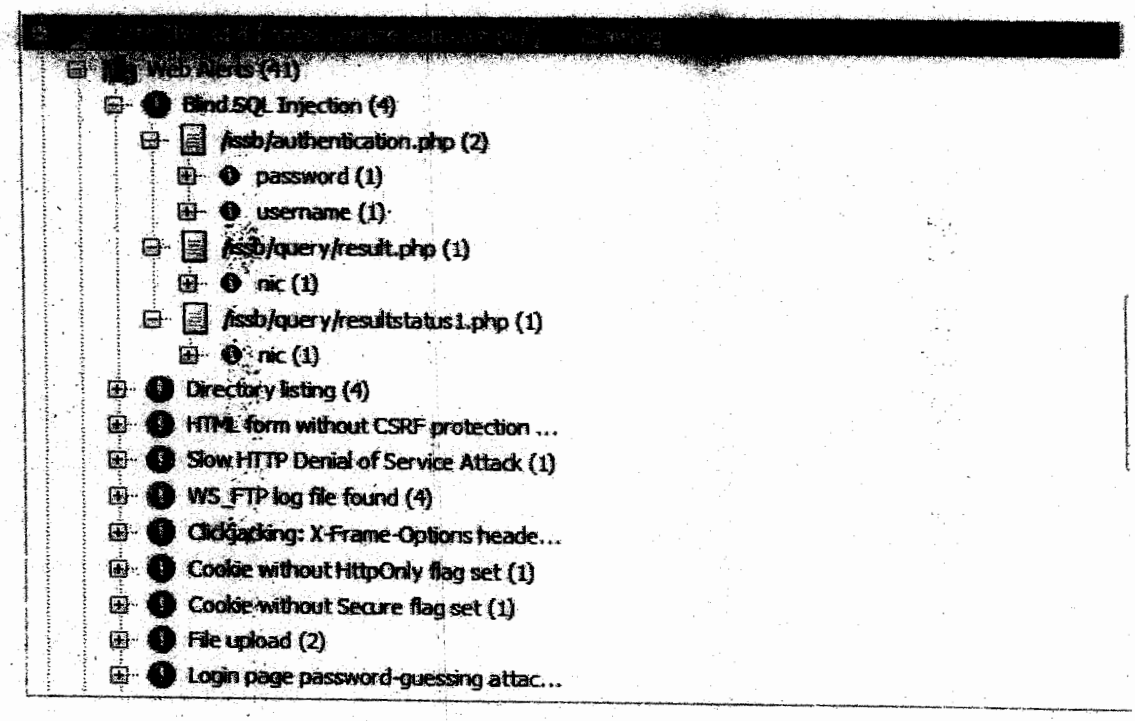
2. Details of vulnerabilities are as under: -

Ser	Vulnerability	Description
a.	Bind SQL Injection (online.issb.com.pk) **(Appex-I)**	Attacker can gain unauthorized access to underlying database and view / manipulate content. Vulnerability can result in reverse shell execution, resulting in remote control over sensitive data leading to leakage and defacement.
b.	Directory Listing Enable **(Appex-II)**	Directory Listing is enabled on website that can results in surface during exploitation. HTML form can compromise user data.

3. Mitigation measures of above referred vulnerabilities and guidelines for prevention against website exploitation are attached at **Appex-III** for compliance, please.

Appex-I

BLIND SQL INJECTION



Appex-II

DIRECTORY LISTING ENABLED

online.issb.com.pk/issb/images/

icon_picture_gallery..>	2013-01-21 07:28	63
icons/	2013-01-21 07:28	-
imagebgoff.gif	2013-01-21 07:28	177
logo.jpg	2013-01-21 07:28	4.0K
navhover.png	2013-01-21 07:28	219
navoff.png	2013-01-21 07:28	219
no.jpg	2013-01-21 07:28	762
picture_middle_crick..>	2013-01-21 07:28	9.5K
point_2.jpg	2013-01-21 07:28	388
print.gif	2013-01-21 07:28	109
sample2_arrow.gif	2013-01-21 07:28	128
sample2_bullet.gif	2013-01-21 07:28	183
sample2_bullet_hl.gif	2013-01-21 07:28	185
space.gif	2013-01-21 07:28	43
spacer.gif	2013-01-21 07:28	71
top_lef.gif	2013-01-21 07:28	952
top_mid.gif	2013-01-21 07:28	86
top_rig.gif	2013-01-21 07:28	961
yes.jpg	2013-01-21 07:28	838

GUIDELINES FOR PREVENTION AGAINST WEBSITE EXPLOITATION

1. **Prevention against SQL Injection**

   a. **Input Validation.** Data that is received from external parties must be validated such that only the value that passes the validation can be processed. It helps counteract any commands inserted in the input string.

   b. **Use of Parameterized Queries.** By employing parameterized queries, user input is automatically quoted and the user / attacker supplied input will not cause the change of the intent. This coding style helps prevent SQL injection attack.

   c. **Use of Stored Procedures.** Stored procedures can reduce direct access to fractions of database, making it an essential in database security.

   d. **Escaping.** Always use character-escaping functions for user-supplied input provided by each database management system (DBMS). This is done to make sure that DMBS never confuses it with SQL statement provided by developer. For example, use mysql_real_escape_string() in PHP to avoid characters that could lead to an unintended SQL command.

   e. **Avoiding Administrative Privileges.** Application must not be connected to the database using an account with root access. This should be done only if absolutely necessary as the attackers could gain access to the whole system.

2. **Prevention against Directory Listing** Web servers should be configured to disable directory listing by default.

3. **General Security Measures**

   a. Upgrade OS and webserver to latest version.

   b. Proper verification of valid ONIC and phone number of users, as per the information in Database must be ensured at the time of registration.

   c. Two factor authentication must be ensured by sending an OTP to registered phone number only, in order to avoid any illegitimate / unauthorized access to potentially sensitive data.

   d. Website admin panel should only be accessible via white-listed IPs.

   e. Defend your website against DQL injection attacks by using input validation technique.

   f. Complete analysis and penetration testing of application be carried out to identify potential threats.

   g. Complete website be deployed on inland servers including database and web infrastructure.

h. HTTPS protocol be used for communication between client and web server.

i. Application and database be installed on different machines with proper security hardening

j. Sensitive data be stored in encrypted from with no direct public access.

k. DB users privileges be minimized and limited access be granted inside programming code

l. Proper security hardening of endpoints and servers be performed and no unnecessary ports and applications be used.

m. Updated Antivirus tools / Firewalls be used on both endpoints and servers to safeguard from potential threats.

n. Enforce a strong password usage policy.

o. Remote management services like RDP and SSH must be disabled in production environment.

p. Deploy web application firewalls for protection against web attacks.

q. Employ secure coding practices such as parameterized queries and proper input sanitization and validation to remove malicious scripts.

r. Avoid using insecure methods like print stack trace in production environment which can disclose important information.

s. Keep system and network devices up to date.

t. Log retention policy must be devised for at least 3x months on separate device, for attacker's reconnaissance.