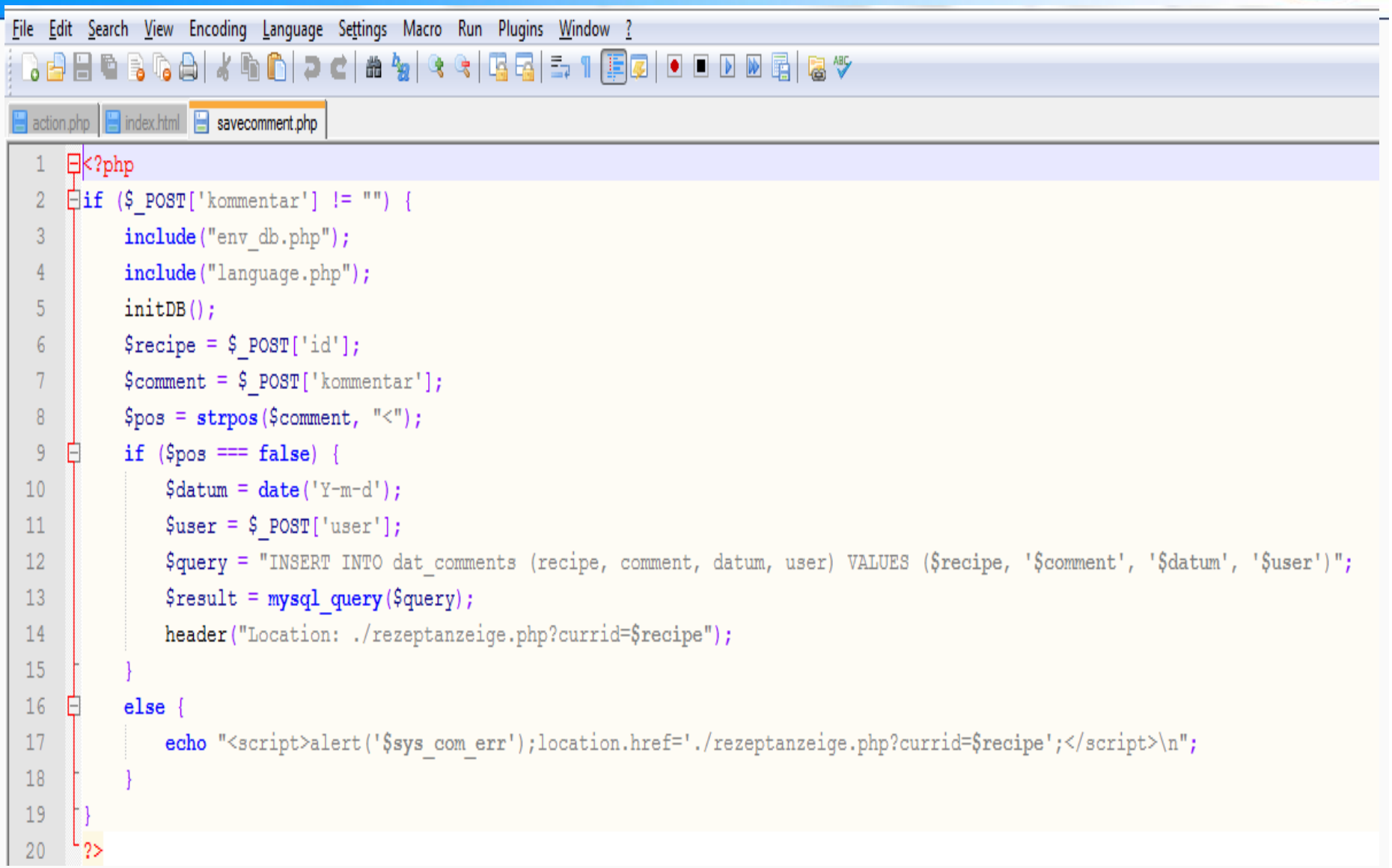
Secure Coding

Shruti Kalsi

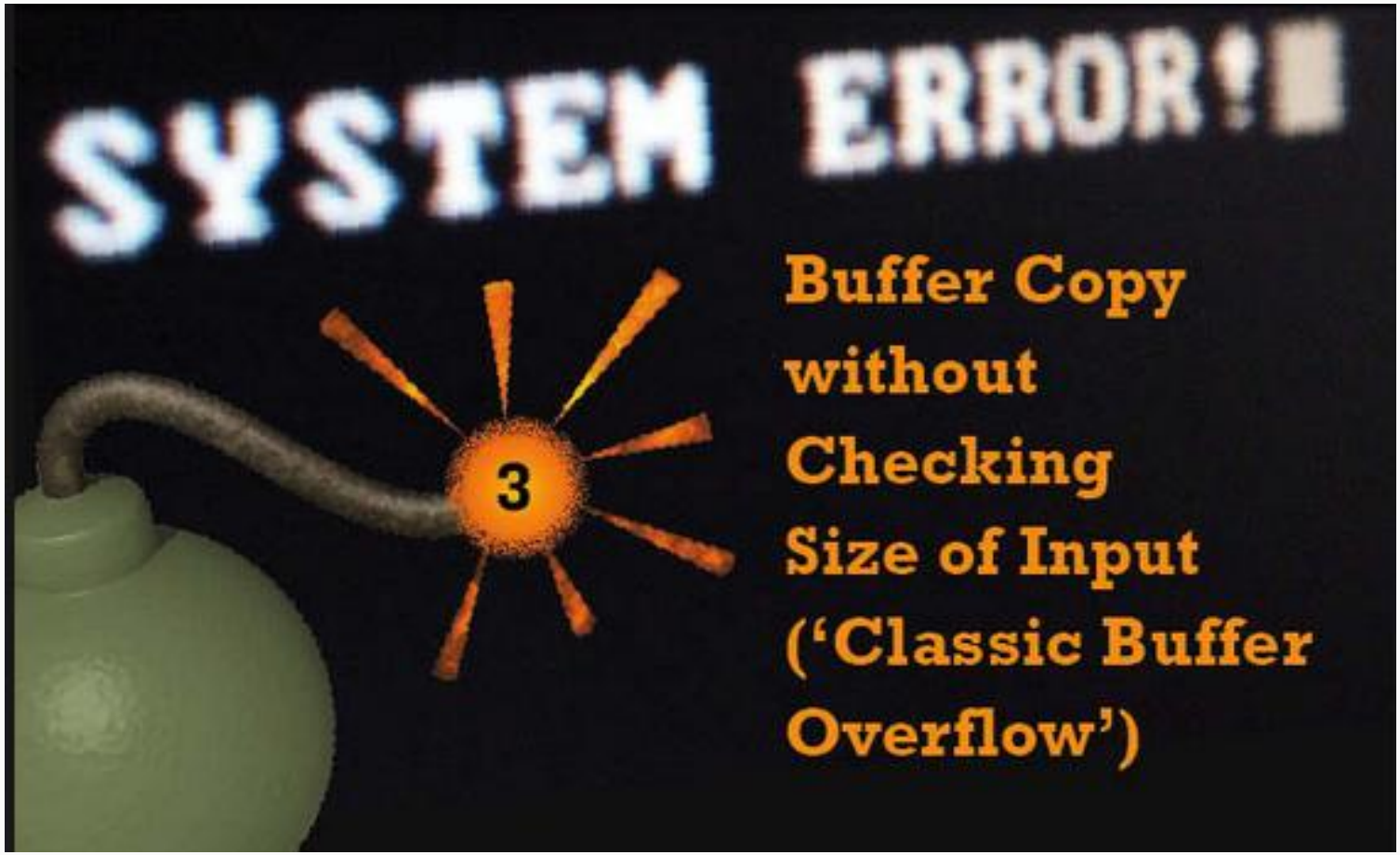
Consultant

Indian Computer emergency Response Team



An illustration of <PHP> source code

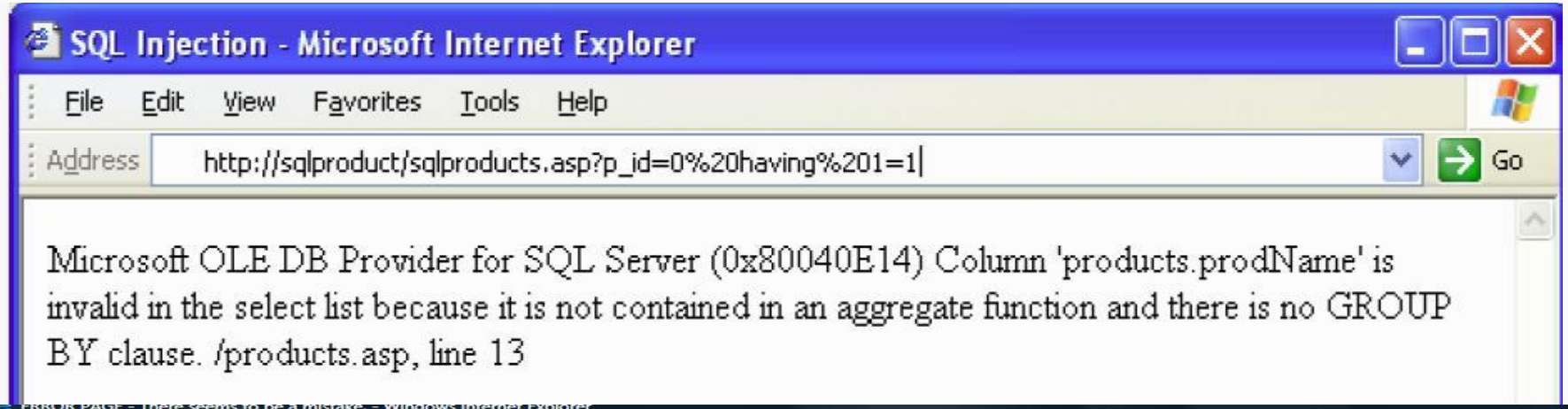
SANS TOP 25 Programming Errors

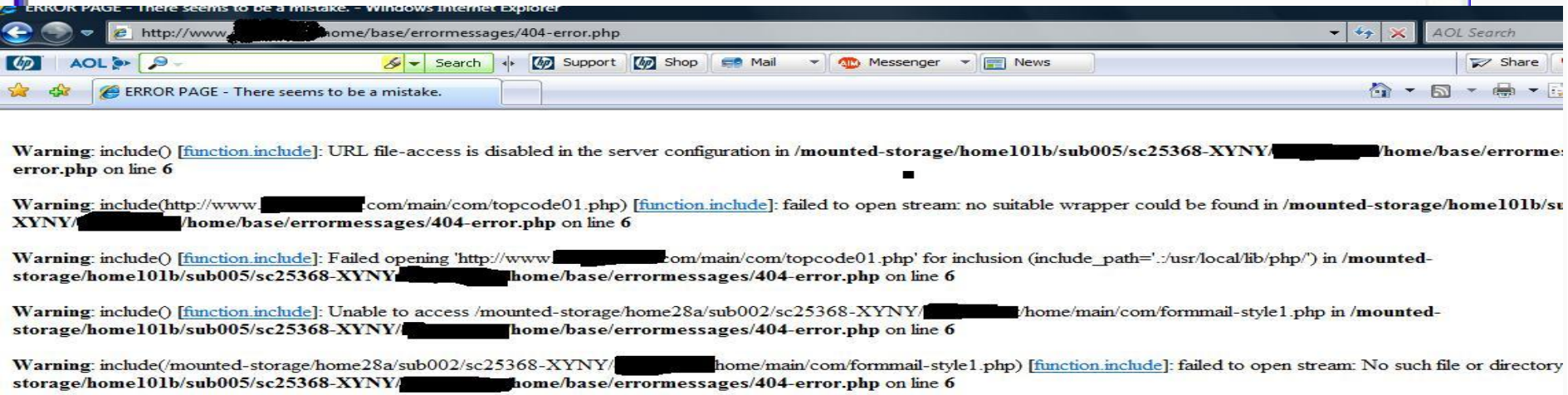


- **Insecure Interaction Between Components**
- **Risky Resource Management**
- **Porous Defences**

- **Insecure Interaction Between Components**
- SQL Injection
- OS Command Injection
- Cross-site Scripting
- Unrestricted Upload of File
- Cross-Site Request Forgery (CSRF)
- URL Redirection to Untrusted Site ('Open Redirect')

Error messages





- **Risky Resource Management**
- Buffer Overflow
- Path Traversal
- Download of Code Without Integrity Check
- Inclusion of Functionality from Untrusted Control Sphere
- Use of Potentially Dangerous Function
- Incorrect Calculation of Buffer Size
- Integer Overflow or Wraparound

- **Porous Defences**
- Missing Authentication for Critical Function
- Missing Authorization
- Use of Hard-coded Credentials
- Missing Encryption of Sensitive Data
- Reliance on Untrusted Inputs in a Security Decision
- Execution with Unnecessary Privileges
- Incorrect Authorization
- Incorrect Permission Assignment for Critical Resource

- **Porous Defences (cont..)**
- Use of a Broken or Risky Cryptographic Algorithm
- Improper Restriction of Excessive Authentication Attempts

Secure Coding Practices.

Input Handling

- Conduct all data validation on a trusted system.
- Validate all input against a "white" list of allowed characters, whenever possible.
- Validate data from redirects.

Escaping

- Using escape Functions/strings for escaping certain characters.

Safe HTML(Sanitization)

- Allowing/Disallowing certain HTML Tags.

File Handling

- Do not pass user supplied data directly to any dynamic include function
- Require authentication before allowing a file to be uploaded
- Limit the type of files that can be uploaded to only those types that are needed
- Turn off execution privileges on file upload directories

Cryptographic Practices

- All cryptographic functions used to protect secrets from the application user must be implemented on a trusted system.
- All random numbers, random file names using the cryptographic module's approved random number generator when these random values are intended to be un-guessable .

Error Handling & Exceptions

- Do not disclose sensitive information in error responses.
- Error handling logic associated with security controls should deny access by default
- Log all input validation failures
- Log all authentication attempts, especially failures

Authenticity Validation

- Who is a valid user to access the website.

Resources for the developers- OWASP, PHPSC.

- PHP Security Consortium (PHPSC) is an international group of PHP experts dedicated to promoting secure programming practices within the PHP community.

References

- http://www.sans.org
- www.securecoding.cert.org
- http://phpsec.org/

Security is a path, not a destination