

# Programming Security Checklist

## Security Review Questions – Authentication

Authentication section covers attacks that target a web site's method of validating the identity of a user, service or application. Authentication is performed using at least one of three mechanisms: "*something you have*", "*something you know*" or "*something you are*".

1. How do users log into the system?
2. What type of information is contained in the system (excluding login Id and password); is it considered "Internal Use Only", "Confidential", or "Secret"?
3. What type of users use this system?  
E.g., External customers, internal users, "Trusted users" General public... (Please specify)
4. What are the Password rules? (E.g., Number of characters, Complex Composition)
5. Are Passwords stored in a HASH format in the database?
6. Is the account locked in the DB if there are between 3 and X bad attempts to login to the account?
7. GET, POST & Encryption: Are you only ever sending the ID & Password via the POST method?  
*Ensure that **GET is not used** to send sensitive data as the information is logged in **clear text** even if SSL is used. SSL only encrypts data in transit – not at the destination point. If POST is used the HTTP body is not logged. However the POST method still sends data as clear text, thus **encryption is vital**.*  
*Ensure that encryption is used for sensitive data at the application level.*
8. Are you **ever** passing the user's password in the clear over the network internally?  
(Is the user's password being transmitted in plaintext - internally, externally, or both?)
9. Are you storing the user's password anywhere other than the database?  
(In hidden field/cookie/session object?)  
If yes, where? (Please specify)

## **Security Review Questions – Authorization**

Authorization section covers attacks that target a web site's method of determining if a user, service, or application has the necessary permissions to perform a requested action.

1. How is it ensured and enforced that users are only allowed to see data that *they* have been authorized to view? (Please specify)
2. How granular is your Authorization check? (Answer the questions below)
3. Does it support access levels? e.g. user, group, admin, etc.
4. Does it support permissions per access level? e.g. read, write, delete, create, etc.
5. Does every page have an explicit check to your authorization logic?

*A check is made on every request received from the browser.*

## **Cookie/Session Management**

Credential/Session Prediction is a method of hijacking or impersonating a web site user. Deducing or guessing the unique value that identifies a particular session or user accomplishes the attack. Also known as Session Hijacking, the consequences could allow attackers the ability to issue web site requests with the compromised user's privileges.

1. Is **encryption** enabled for all confidential or highly confidential information that is being passed in cookies?
2. If the application creates its own session identifier, what method(s) were used to ensure it is random & unique?
3. What methods are used to ensure your sessions expire both after disuse (HTTP timeout) as well as after a set length of time (Hardlimit)?
4. Is there a session validation strategy? If so, please describe when is the user's session validated (ideally should be done on every request received)? *(please specify)*

## ***Security Review Questions – Data Validation***

1. Is there validation being performed for every source of input from the browser that comes into the application? (Answer questions below)
2. All form data (text boxes, select boxes, hidden fields, etc.)
3. Cookies used by your application
4. Headers (Things such as REFERER, USERAGENT, Content-Length, Content-Type etc.)
5. What type of validation rules does the application use?
6. Is the strongest level of validation possible being used?
7. Has the maximum length for each field been checked?
8. If data fails validation for being too long, is this information logged in its entirety?
9. Is bad data encoded before echoing it back to the browser to prevent cross-site scripting issues?

## ***Error Handling***

1. Is there an Error handling strategy, if so, please describe?
2. Is logic built into the application to 'fail securely'?  
(Basically establishing a default fail stance, and only letting something continue when it has explicitly passed?)

## ***Security Review Questions – Logging/Debugging/Error Messages***

1. Have steps been taken to ensure against logging any confidential/secret information, such as password?
2. Have steps been taken to ensure only business error messages (such as 'you do not have rights to that function') are being sent back to the browser?
3. Have steps been taken to ensure the main error page for serious system problems is generic and not too revealing of the problem that occurred on the server?
4. Have steps been taken to ensure the logging strategy recognizes the following?
  - User input you have logged, possibly as a result of an error, is clean of things that might cause a cross-site scripting attack if later viewed through a browser?
  - Logging user input without length checks?

## PIM Background

### *Personal Information Defined*

The term “Personal Information” shall mean any and all information pertaining to a specific person, including without limitation, a person’s e-mail address, name, mailing address, telephone number, social security number, credit card number, or persistent identifier (such as a customer number held in a cookie) which is associated with a person’s individually identifiable information, which is (A) disclosed or furnished, in any form, by the company, its affiliates, or agents or (B) collected on behalf of the company by a vendor in connection with a vendor’s performance of the Services.

Personal Information also includes information that can, together with the other information supplied by the company or collected or to be collected by a vendor, identify a specific individual, even if such information cannot, by itself, identify a specific individual.

### *PIM Logging*

Keep all logs used in connection with its security procedures for the protection of **Personal Information** in a secure location for a period beginning as of the Effective Date and ending [two (2)] years after the termination of any agreement with the customer, client, or vendor.

## PCI Background

### *Payment Card Industry (PCI) Defined*

The PCI DSS version 1.1, a set of comprehensive requirements for enhancing payment account data security, was developed by the founding payment brands of the PCI Security Standards Council, including American Express, Discover Financial Services, JCB International, MasterCard Worldwide and Visa Inc., to help facilitate the broad adoption of consistent data security measures on a global basis.

The core of the PCI DSS is a group of principles and accompanying requirements, around which the specific elements of the DSS are organized:

#### **Build and Maintain a Secure Network**

*Requirement 1:* Install and maintain a firewall configuration to protect cardholder data

*Requirement 2:* Do not use vendor-supplied defaults for system passwords and other security parameters

#### **Protect Cardholder Data**

*Requirement 3:* [Protect stored cardholder data](#)

*Requirement 4:* Encrypt transmission of cardholder data across open, public networks

#### **Maintain a Vulnerability Management Program**

*Requirement 5:* Use and regularly update anti-virus software

*Requirement 6:* Develop and maintain secure systems and applications

#### **Implement Strong Access Control Measures**

*Requirement 7:* Restrict access to cardholder data by business need-to-know

*Requirement 8:* Assign a unique ID to each person with computer access

*Requirement 9:* Restrict physical access to cardholder data

#### **Regularly Monitor and Test Networks**

*Requirement 10:* Track and monitor all access to network resources and cardholder data

*Requirement 11:* Regularly test security systems and processes

#### **Maintain an Information Security Policy**

*Requirement 12:* Maintain a policy that addresses information security

## Requirement 3: Protect stored cardholder data

*Encryption is a critical component of cardholder data protection. If an intruder circumvents other network security controls and gains access to encrypted data, without the proper cryptographic keys, the data is unreadable and unusable to that person. Other effective methods of protecting stored data should be considered as potential risk mitigation opportunities. For example, methods for minimizing risk include not storing cardholder data unless absolutely necessary, truncating cardholder data if full PAN is not needed, and not sending PAN in unencrypted e-mails.*

**3.1** Keep cardholder data storage to a minimum. Develop a data retention and disposal policy. Limit storage amount and **retention time** to that which is required for business, legal, and/or regulatory purposes, as documented in the data retention policy.

**3.2** Do not store sensitive authentication data **subsequent to authorization** (even if encrypted). Sensitive authentication data includes the data as cited in the following Requirements 3.2.1 through 3.2.3:

**3.2.1** Do not store the full contents of any track from the magnetic stripe (that is on the back of a card, in a chip or elsewhere). This data is alternatively called full track, track, track 1, track 2, and magnetic stripe data. In the normal course of business, the following data elements from the magnetic stripe may need to be retained: the account holder's name, primary account number (PAN), expiration date, and service code. To minimize risk, **store only those data elements needed for business**. **NEVER** store the card verification code or value or PIN verification value data elements.

**3.2.2** Do not store the card-validation code or value (three-digit or four-digit number printed on the front or back of a payment card) used to verify card-not-present transactions

**3.2.3** Do not store the personal identification number (PIN) or the encrypted PIN block.

**3.3** **Mask PAN when displayed** (the first six and last four digits are the maximum number of digits to be displayed).

*Note: This requirement does not apply to employees and other parties with a specific need to see the full PAN; nor does the requirement supersede stricter requirements in place for displays of cardholder data (for example, for point of sale [POS] receipts).*

**3.4** Render PAN, at minimum, unreadable anywhere it is stored (including data on portable digital media, backup media, in logs, and data received from or stored by wireless networks) by using any of the following approaches:

- Strong one-way hash functions (hashed indexes)
- Truncation
- Index tokens and pads (pads must be securely stored)
- Strong cryptography with associated key management processes and procedures.

**The MINIMUM account information that must be rendered unreadable is the PAN.**

If for some reason, a company is unable to encrypt cardholder data, refer to Appendix B:

“Compensating Controls for Encryption of Stored Data.”

**3.4.1** If disk encryption is used (rather than file- or column-level database encryption), logical access must be managed independently of native operating system access control mechanisms (for example, by not using local system or Active Directory accounts). Decryption keys must not be tied to user accounts.

**3.5** Protect encryption keys used for encryption of cardholder data against both disclosure and misuse.

**3.5.1** Restrict access to keys to the fewest number of custodians necessary

**3.5.2** Store keys securely in the fewest possible locations and forms.

**3.6** Fully document and implement all key management processes and procedures for keys used for encryption of cardholder data, including the following:

**3.6.1** Generation of strong keys

**3.6.2** Secure key distribution

**3.6.3** Secure key storage

**3.6.4** Periodic changing of keys

- As deemed necessary and recommended by the associated application (for example, re-keying); preferably automatically

- At least annually.

**3.6.5** Destruction of old keys

**3.6.6** Split knowledge and establishment of dual control of keys (so that it requires two or three people, each knowing only their part of the key, to reconstruct the whole key)

**3.6.7** Prevention of unauthorized substitution of keys

**3.6.8** Replacement of known or suspected compromised keys

**3.6.9** Revocation of old or invalid keys

**3.6.10** Requirement for key custodians to sign a form stating that they understand and accept their key-custodian responsibilities.

## Cryptographic technologies

Cryptographic technologies used shall meet the following criteria:

- When using protocols implementing cryptographic technology, de facto standard protocols shall be used. (e.g. SSL, SSH, IPSec, S/MIME etc)
- Proven, standard algorithms shall be used as the basis for encryption technologies. (e.g. AES, ECDSA, ECDH, RSA-PSS, SHA-256, MISTY1, Camellia, 3DES etc.)
- The length of the key and the message digest shall have strength equivalent to or greater than what is required in order to be secure under current technology.

The following are **guidelines** for required lengths:

- (i) Symmetric cryptosystem key lengths shall be **at least 40 bits**.
  - (ii) Asymmetric cryptosystem keys shall be of a length equivalent to or **more than the strength of 512 bits** for the RSA algorithm.
  - (iii) A minimum of 128 bits shall be used for the message digest.
- Along with the algorithm, key length and the length of the message digest shall be reviewed periodically and enhanced as necessary.

- Due to the nature of encryption technology, key and/or data recovery logic shall be implemented and documented to ensure the recoverability of enciphered data. Also, a framework for detecting and avoiding any unauthorized recovery of encrypted data shall be established.
- For implementations where data encryption is used, a framework and process shall be established so that the data can be quickly and easily restored whenever needed.
- Project plans for application development shall include encryption and cryptographic requirements in the design phase of each project.
- All encryption or cryptographic implementations shall include a recovery mechanism to allow for a forensic investigation. Keys must be readily available upon request by law enforcement.