Ασφάλεια δεδομένων και συναλλαγών

XSS

Το πρόβλημα

Κατά την επίθεση τύπου XSS ο επιτιθέμενος επιδιώκει την εισαγωγή κακόβουλου κώδικα σε μια σελίδα εκμεταλλευόμενος την αδυναμία ελέγχου των δεδομένων που λαμβάνει η σελίδα και ταυτόχρονα την αδυναμία ελέγχου των δεδομένων πριν την εμφάνισή τους σε χρήστες.

Η λύση

Θα πρέπει να χρησιμοποιούνται τόσο οι μέθοδοι καθαρισμού δεδομένων όπως η strip_tags() της PHP που μας βοηθάει να αποφεύγουμε ορισμένες ετικέτες (όπως είναι η <script>) κατά τη λήψη των δεδομένων μιας φόρμας, όσο και οι μέθοδοι αντικατάστασης επικίνδυνων συμβόλων htmlspecialchars και htmlentities κατά την εμφάνιση δεδομένων.

Χρήσιμοι σύνδεσμοι

- http://php.net/manual/en/function.strip-tags.php
- http://php.net/manual/en/function.htmlspecialchars.php
- http://php.net/manual/en/function.htmlentities.php
- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

CSRF

Το πρόβλημα

Κατά την επίθεση τύπου CSRF ο χρήστης αναγκάζεται να εκτελέσει ανεπιθύμητες ενέργειες σε μια ιστοσελίδα που είναι συνδεδεμένος.

Η λύση

Θα πρέπει να χρησιμοποιούνται tokens για τον έλεγχο της προέλευσης κάθε αιτήματος προς τον web server.

Χρήσιμοι σύνδεσμοι

- https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)

SQL injection

Το πρόβλημα

Κατά την επίθεση τύπου SQL injection ο επιτιθέμενος καταφέρνει να εισάγει στα ερωτήματα προς τη βάση δεδομένων κακόβουλο κώδικα.

Η λύση

Θα πρέπει να χρησιμοποιούνται οι μέθοδοι διαφυγής (escape functions) όπως η mysqli::real_escape_string(), καθώς και προεπεξεργασμένα ερωτήματα (prepared statements).

Χρήσιμοι σύνδεσμοι

- http://php.net/manual/en/mysqli.real-escape-string.php
- http://php.net/manual/en/mysqli.prepare.php
- http://php.net/manual/en/mysqli.quickstart.prepared-statements.php
- https://www.owasp.org/index.php/SQL_Injection

Άλλοι τύποι επιθέσεων

https://www.owasp.org/index.php/Category:Attack