הנחיות לפיתוח מאובטח

אוניברסיטת חיפה

בוצע ע"י:

אגף מחשוב ומערכות מידע

מרץ 2018

תוכן עיניינים

ניהול האתר

במידה וניהול המשתמשים/סיסמאות/לוגים/הרשאות מתבצע במערכת ולא בשרת חובה לפעול לפי מסמך הנחיות לניהול אתר web, ונוהל סיסמאות אוניברסיטאי.

הנחיות בנוגע לכתיבה מוקשחת

אימות קלט

יש לוודא הגנה על האותנטיות של פרמטרים, ולבצע בדיקת קלט קפדניות. ניסיון לשנות פרמטרים שמועברים לשרת הם הבסיס לתקיפות מסוג SQL Injection, Cross Site scripting ודומיהם.
בדיקות ואימות הקלט יתבצעו גם בצד השרת וגם בצד הלקוח ויכללו בין היתר:

- ☒ במסגרת בדיקת אורך הקלט, יש לוודא שימוש רק תווים מותרים "white list".
- ☒ יש להימנע ממתן יכולת "free format input" ולהגדיר הגבלות ככל שניתן, לדוגמא יש להעדיף בחירה מתפריט לעומת תיבת טקסט.

- ☒ על אף שלטובת חווית המשתמש אימות הקלט יעשה בצד המשתמש, אין להסתמך על בדיקות בצד לקוח.
- ☒ לא להשתמש בערכים שהתקבלו ישירות מהמשתמש לצורך יצירת דף דינאמי.
- ☒ אין לאפשר שאילתות דינאמיות, במקומן יש להשתמש בפרוצדורות. השימוש בפרוצדורות מוכנות עדיף על שימוש בשאילתות דינאמיות, מאחר שלא ניתן להשפיע עליהן באופן זדוני. למידע ודוגמאות, מצורף לינק:

http://php.net/manual/en/mysqli.quickstart.stored-procedures.ph

הגנה על מידע רגיש

- ☒ על מנת להמנע למלהציג את הסיסמא במערכות בקרה שונות יש להקפיד להציב את הסיסמא במשתנה מסוג סיסמא.
- ☒ יש להצפין נתונים רגישים במערכת (הן בקבצים והן בבסיס הנתונים), במיוחד UserID & Password.
- ☒ מערך ההרשאות הגישה ל Database יהיה לפי מדיניות של הרשאות מינימליות עבור תהליך/משתמש.
- ☒ יש להקפיד על הפרדה של משתמשים, משתמש יעודי לכל תהליך
- ☒ להמנע למהציג כל נתוני זיהוי של רכיבי תוכנה כלשהם, דוגמת מידע נתוני הזיהוי של שרת האפליקציה לשרת בסיס הנתונים. דוגמה banner של שרת web, ftp.
- ☒ מפתח ההצפנה יישמר במקום מאובטח על שרת המערכת, כגון ה- Registry (הגישה למפתח תוגבל לאפליקציה ולאדמיניסטראטור של השרת בלבד)
- ☒ יש לשמור עותק של המפתח במקום מוגן נפרד (פיזי) למקרה שלא ניתן לשחזר את המפתח המקורי.
- ☒ יש להשתמש בהצפנות מקובלות כיום בשוק, כגון RSA, ולא לבנות אלגוריתם הצפנה ייחודי למערכת.
- ☒ אין לאפשר שמירת נתונים רגישים של המערכת במחשבו של המשתמש.
- ☒ יש להימנע משמירת נתוני המערכת בספריית הקבצים הזמניים ובמנגנוני Caches של מחשב המשתמש.

דגשים בהליך הפיתוח

- ☒ יש לוודא שכל הסיפריות ו/או הרכיבים בהן נעשה שימוש בפיתוח ילקחו ממקורות מוכרים ובטוחים וכן נקיים מחולשות אבט״מ ידועות נכון למועד מסירת המוצר.
- ☒ מערכת הכוללת מודול הזדהות עצמאי, אזי על מנת להתמודד עם התקפה מסוג של מניעת שירות – DOS Attacks המערכת תכלול רכיב כמו CAPTCHA כאשר לאחר מספר מוגדר של כישלונות המשתמש יחסם לפרק זמן של 15 דקות.

ניהול Session Management

- ☒ יש להבטיח כי נתוני session נשמרים בצורה בטוחה.
- ☒ יש להבטיח כי קיימת הפרדה בין ניהול הזהויות לבין שימוש ב session . החשש שהוא מתהליך של גניבת זהות, מצב שמשתמש שלא ביצע הזדהות יוכל להשתמש ב session פעיל של משתמש אחר שביצע הזדהות כנדרש. במילים אחרות, יש להבטיח כי המערכת אינה מסתמכת על נתוני session בכדי לאפשר למשתמש חשיפה למידע ופעולות רגישים במערכת.
- ☒ יש להשתמש ברכיבי session רק עבור שמירת מצב משתמש בין בקשות http שונות במערכת וכן לצורך ביצוע personalization עבור משתמש.
- ☒ אין לשמור מידע רגיש ב- session , במידה ונדרש יש לבצע הצפנה של מידע זה.
- ☒ בכל מצב שבו נשמר מידע רגיש ב session יש להבטיח כי המידע נשמר בצורה בטוחה ולא תתאפשר גישה אליו שלא דרך מקור מוסמך ומאושר (כלומר מהאפליקציה שייצרה את המידע).
- ☒ האפליקציה תעשה שימוש רק בזהות אשר נתקבלה בתהליך ההזדהות בכניסה לאפליקציה ואשר מבצעת שימוש ב- Session ID ייחודי וזמני.
- ☒ יש למנוע ביצוע גישה למערכת ללא session תקין.
- ☒ אין להעביר את נתוני הזיהוי של המשתמשים בין מחשב המשתמש לשרתי המערכת, למעט דף הכניסה למערכת.
- ☒ Session לא סגור יכול להיות פתח לגניבת זהותו של המשתמש המקורי. לפיכך, יש לקיים מנגנון Idle Timeout אשר יסיים את Sessions של המשתמש לאחר מספר דקות מוגדר, לדוגמא כ-15 דקות, של חוסר פעילות במערכת.
- ☒ ניתוק Sessions יבוצע על ידי סיום תוקף Sessions בצד השרת, ולא על ידי העברת הלקוח לדף הכניסה בלבד.

הגנה מפני מתקפות אפליקטיביות

מניעת התקפות cross site scripting

יש לבצע בדיקות תקינות בצד השרת על כל הקלט המגיע מצד המשתמש. בדיקת הקלט תכלול את הבדיקות הבאות:

- ☒ יש לבדוק את קיומו של הקלט ולא לאפשר הזנת ערכים ריקים.
- ☒ יש לבדוק ולהגביל את אורך הקלט.
- ☒ יש לבדוק שסוג הקלט המתקבל הוא אכן מהסוג המצופה.
- ☒ יש לבדוק כי טווח הערכים שמתקבל מתאים להגבלות שנקבעו.
- ☒ יש לבדוק את הרכב התווים בקלט, ולוודא שהוא אינו מכיל תווים אסורים.
- ☒ יש לוודא כי הקלט ב encoding מתאים למערכת.

- ☒ יש להעביר את כלל התווים שאינם אלפאנומריים קידוד HTML בטרם הצגתם למשתמש. תהליך הקידוד יבטיח כי קוד שתול יוצג כטקסט ולא ירוץ על הדפדפן.
- ☒ אין להכניס לבסיס הנתונים תווים הנובעים מקלט ישירות לתחום הפעולה של client side scripting (תגי script, אירועי HTML וכדומה).

מניעת הזרקות SQL

- ☒ אין לאפשר גישה ישירה לבסיס הנתונים. גישה לבסיס הנתונים תתבצע באמצעות שיכבה מתווכת כגון WS או DAL (Data Access Layer).
- ☒ בכל מקרה יש לבצע סינון מסודר של תווים למניעת הזרקת שאילתות SQL
- ☒ כל תעבורת השאילתות תבוצע על ידי שימוש ב- stored procedures באופן הנכון וללא שימוש בהעברת פרמטרים בקריאה ל- stored procedure.

הגנה מפני buffer overflow

- ☒ יש לאמת פרמטרי מחרוזות כקלט ופלט, לשם כך יש לוודא שאורך המחרוזת שלא תחרוג מהמקסימום.
- ☒ יש לאמת גבולות של מערכים.
- ☒ יש לאמת אורך נתיב לקבצים.

הגנה מפני מתקפות Network eavesdropping

- ☒ יש להצפין את תווך תקשורת.

הגנה מפני מתקפות Brute force & Dictionary attacks

- ☒ יש להקפיד על מימוש מדיניות סיסמאות לפי מסמך מדיניות סיסמאות.
- ☒ שמירת סיסמאות יתבצע על ידי שימוש ב hash בתוספת מספר רנדומאלי.
- ☒ יש להימנע ככל האפשר מפתיחה של יותר מ session אחד עבור משתמש
- ☒ יש לבצע בדיקות אימות ל- session לפני מתן גישה כלשהיא.
- ☒ יש להשתמש בתווך מוצפן תוך שימוש באלגוריתם הצפנה חזק בכדי שלא ניתן יהיה לגנוב cookie המועבר לאפליקציה.
- ☒ לשם מניעת מתקפות replay, יש ליצור ערך חד ערכי עבור כל הודעה נשלחת. כמו כן, מומלץ לשלב חתימה בגוף ההודעה - timestamp.
- ☒ יש להימנע משמירת נתונים במטמון הדפדפן

- ☒ אופציית אחסון הדפים (Caching) תהיה מבוטלת עבור כל הדפים באפליקציה ולכל סוגי הדפדפנים.
- ☒ יש לבטל אפשרות ה- Password Auto complete על ידי שליחת מאפיין מתאים בתגי Form-m בדף HTML, לדוגמא: Passwords

<"INPUT TYPE="password" AUTOCOMPLETE= "off>

הגנה מפני מתקפות CSRF:

- ☒ הגבלת זמן השהייה בחשבון - פקיעת תוקף לאחר זמן מסוים בו לא בוצעה שום פעולה.
- ☒ יש לבקש אישור מהמשתמש בעבור כל פעולה חשובה המתבצעת באתר.
- ☒ בעבור כל שליחת טופס ניתן להוסיף שדה נסתר המכיל מספר פסאודו-אקראי אשר יוגש עם הבקשה לביצוע פעולה, ואם הערך לא יוצג בטופס, השרת יתעלם מהפעולה.
- ☒ אימות הכותרת המפנה (referrer header), אשר מבטיח כי הערך זמין רק עבור סקריפטים מהדף המקורי.
- ☒ שליחת בקשות בשיטת POST ולא באמצעות GET.

מקורות נוספים:

- SANS - A Security Checklist for Web Application Design
- The Security Checklist
- Web Developer Security Checklist - Michael O' Brien's