**Dor Erlich 19.05.2024**

**הגנה מפני התקפות XSS (Cross-Site Scripting) בחברות**

**מבוא**

התקפות XSS (Cross-Site Scripting) הן סוג של פגיעות אבטחה באפליקציות ווב, בהן תוקף מצליח להזריק קוד זדוני לדפדפן המשתמש. התקפות אלו עלולות לגרום לגניבת מידע אישי, התחזות למשתמשים, והפצת תוכנות זדוניות. מטרת מאמר זה היא להסביר את סוגי התקפות XSS, דרכי ההתגוננות מהן, ואמצעי מניעה שיכולים לסייע בהגנה על אפליקציות ווב של חברות.

**סוגי תקיפות XSS**

התקפת XSS מאוחסנת:

בהתקפה זו, הקוד הזדוני נשמר בשרת האפליקציה, למשל במסד נתונים. כאשר משתמש גולש לעמוד שמכיל את הקוד הזה, הדפדפן שלו מריץ את הקוד הזדוני. דוגמה נפוצה לכך היא טופס שבו משתמש מכניס קוד זדוני, והקוד נשמר ומוצג לאחר מכן לכל משתמש שניגש לעמוד זה.

התקפת XSS משתקפת:

בהתקפה זו, הקוד הזדוני נשלח למשתמש כחלק מבקשה (למשל, בקשה עם פרמטרים ב-URL). הדפדפן מריץ את הקוד הזדוני שמגיע עם הבקשה ומציג את התוצאה למשתמש.

התקפת XSS מבוססת DOM:

בהתקפה זו, הקוד הזדוני משנה את מבנה ה-DOM (Document Object Model) בדפדפן של המשתמש, מה שגורם להפעלת הקוד הזדוני בצד הלקוח. התקפה זו אינה תלויה בשרת, אלא מתבצעת באופן ישיר על הדפדפן של המשתמש.

**דרכי התמודדות עם התקפות XSS בחברות**

קידוד (Encoding):

קידוד נתונים הוא צעד חיוני בהגנה מפני התקפות XSS. נתונים שהוזנו על ידי המשתמש צריכים להיות מקודדים לפני שמוצגים בדפדפן. למשל, קידוד HTML, קידוד URL, וקידוד JavaScript יכולים למנוע הזרקת קוד זדוני.

השתמשו בCSP (Content Security Policy):

CSP היא כותרת HTTP המגדירה מדיניות אבטחה בדפדפן. ניתן להגדיר CSP שמונעת הפעלת סקריפטים ממקורות לא מאושרים, ובכך להגביל את היכולת של קוד זדוני להתבצע בדפדפן.

בדיקות אינפוטים (Input Validation):

חשוב לבדוק את כל הנתונים המתקבלים מהמשתמש ולוודא שהם עומדים בקריטריונים מסוימים. בדיקות אלו יכולות למנוע הכנסת קוד זדוני לאפליקציה.

השתמשו בכלים לזיהוי פגיעויות (Vulnerability Scanners):

קיימים כלים אוטומטיים שיכולים לסרוק את הקוד שלכם ולזהות פגיעויות XSS. כלים אלו יכולים לסייע בזיהוי ותיקון בעיות אבטחה לפני שהן נחשפות לציבור.

**אמצעי מניעה עבור חברות**

הפרדת נתונים מקוד:

הפרדה בין נתונים לקוד יכולה למנוע אפשרות של הזרקת קוד זדוני. שימוש בפרמטרים מופרדים בבקשות SQL, למשל, יכול למנוע התקפות SQL Injection ואחרות.

עדכוני תוכנה תכופים:

שמירה על עדכוני תוכנה ותיקוני אבטחה היא קריטית. עדכונים אלו כוללים לרוב תיקונים לפגיעויות אבטחה שעלולות להיות מנוצלות על ידי תוקפים.

הדרכות אבטחה לצוות הפיתוח:

הכשרת צוות הפיתוח בנושאי אבטחת מידע יכולה למנוע טעויות פיתוח שגורמות לפגיעויות XSS. ידע נכון בנוגע לאבטחת מידע יכול לסייע במניעת בעיות אלו מהשורש.

שימוש ב-frameworks מאובטחים:

frameworks מאובטחים כוללים מנגנוני אבטחה מובנים שיכולים להקל על הגנה מפני התקפות XSS. למשל, frameworks כמו Angular ו-React כוללים מנגנונים לקידוד נתונים ולמניעת הזרקת קוד.

בדיקות אבטחה תקופתיות:

ביצוע בדיקות אבטחה תקופתיות יכול לחשוף נקודות תורפה חדשות שנוצרות עם הזמן. הבדיקות צריכות לכלול בדיקות חדירה (penetration testing) ובדיקות קוד סטטיות.

מנגנוני ניהול לוגים ואנליטיקות:

שימוש במנגנוני ניהול לוגים ואנליטיקות יכול לסייע בזיהוי התקפות בזמן אמת ובתגובה מהירה להן. מנגנונים אלו יכולים לזהות דפוסי פעילות חשודים ולספק התראות למנהלי האבטחה.

**סיכום**

התקפות XSS מהוות איום משמעותי על אבטחת אפליקציות ווב, אך ניתן להתמודד איתן באמצעות שילוב של אמצעי הגנה ומניעה. קידוד נתונים, שימוש ב-CSP, בדיקות אינפוטים, והדרכות אבטחה הם רק חלק מהצעדים שיכולים לסייע בהגנה על המערכת. הקפדה על עקרונות אלו ושימוש בכלים מתאימים תסייע ביצירת סביבה בטוחה יותר למשתמשים ולמערכת כולה.