מסמך הנחיות מפורטות לפיתוח מערכות המנהלות משתמשים

גרסה 0.1

מסמך זה כולל מידע השייך לממשל זמין, רשות התקשוב הממשלתי. כל חשיפה, שימוש או העתקה של מסמך זה או חלקים ממנו – ללא קבלת אישור בכתב ממנהל מערך סייבר ואבטחת מידע בממשל זמין – אסורה בהחלט. מסמך זה מיועד לעובדי ממשל זמין ולקוחותיו

מעקב גרסאות

מס"ד	תאריך	עודכן על ידי	תיאור השינויים
1.0	2.8.2016	יוגב מזרחי	גירסה ראשונה

נתוני גרסת המסמך

גורם	תפקיד	שם מלא	תאריך	חתימה
נערכה ע"י	ראש תחום בדיקות חדירות	יוגב מזרחי	2.8.2016	(חתימה)
נבדקה ע"י				(חתימה)
אושרה ע"י	מנהל מערך הגנה בסייבר	אברהם זרוק	3.8.2016	(חתימה)

תוכן עניינים

1. כללי..... 4
2. הקדמה ..... Error! Bookmark not defined.
3. הנחיות לשמירת סודיות הסיסמאות 4
3.1 מדיניות סיסמאות 4
3.2 שמירה על סודיות הסיסמאות והגנה מפני התקפות..... 4

1. כללי

מסמך זה מהווה קובץ הנחיות מפורטות לפיתוח מערכות המנהלות משתמשים.
במסמך התייחסות מיוחדת לניהול סיסמאות בצורה מאובטחת.
המסמך נכתב בהתאם להנחיות "המכון הלאומי לתקנים וטכנולוגיה" האמריקאי (NIST).
מסמך זה מהווה נספח למסמך מדיניות פיתוח מערכת מאובטחת בממשל זמין.

2. הנחיות לשמירת סודיות הסיסמאות

2.1 מדיניות סיסמאות

2.1.1 יש לשמור על מורכבות הסיסמאות בעת יצירתן בהתאם למדיניות ממשל זמין (מידע נוסף קיים במסמך מדיניות פיתוח מאובטח)

2.2 שמירה על סודיות הסיסמאות והגנה מפני התקפות

2.2.1 אין לשמור קבצים עם סיסמאות גלויות בשרת/ים.

2.2.2 יש להשתמש בהצפנה הסיסמאות ב- HASH חד כיווני.

2.2.3 יש צורך לוודא את זהות המשתמש אשר מבצע תהליך לשחזור או איפוס סיסמה, ולא לאפשר לתוקף לקבל סיסמאות של משתמשים כתוצאה מתהליך כזה או אחר.

2.2.4 יש לשמור על סודיות הסיסמאות לאורך כל הדרך (במנוחה ובתנועה).

2.2.5 במנוחה – יש לשמור על אחסון הסיסמאות בצורה מוצפנת בלבד לרבות בעת שימוש בסיסמאות בקבצי קונפיגורציה.

2.2.6 בתנועה – יש לשמור על העברת סיסמאות בצורה מאובטחת ממקום למקום בכדי למנוע התקפות MITM (Man In the Middle) ולכן יש להעביר סיסמאות באמצעות פרוטוקולים המצפינים את המידע העובר בתווך (למשל העברה באמצעות SSH ולא באמצעות FTP).

2.2.7 יש למנוע התקפות מסוג "Relay Attacks" אשר בהן התוקף הינו באמצע התעבורה ובכך ביכולתו להעביר את הסיסמה המוצפנת כמו שהיא אל מנגנון האימות ובכך אין לו צורך

לבצע תהליך של "פיצוח" הסיסמה לצורך אימות למערכת. ניתן למנוע זאת על ידי שילוב מנגנון האימות עם פונקציות timestamps או עטיפת המנגנון בפרוטוקול המגן על כך כגון TLS.

2.2.8 אין לחשוף את הסיסמה אותה משתמש מזין במסך בכדי למנוע התקפות גניבת סיסמאות מסוג "shoulder surfing", ולכן יש להסתיר את הסיסמה המוזנת לדוגמה באמצעות כוכביות ולא להציגה במסך במהלך תהליך הרשמה או שחזור סיסמה.

2.2.9 ניחוש סיסמאות

2.2.9.1 בכדי למנוע התקפות לניחוש סיסמאות יש למנוע שימוש בסיסמאות קלות לניחוש כגון שימוש בשם המשתמש כסיסמה או כל פרט פומבי אחר הקשור לתהליך ההרשמה, שימוש בסיסמאות של חלשות בצירופי מקלדת כגון qwert, 1234!@#$

2.2.9.2 יש להגביל את מספר הניסיונות להיכנס לחשבון במרווח זמן מסויים על ידי שימוש בטכניקות כגון:

- נעילת משתמש ל-15 דק' לאחר X ניסיונות שנכשלו.
- יצירת השהיית זמן רנדומלי בין ניסיון לניסיון כך שלא יפגע בחווית המשתמש אך כן יגן על מנגנון האימות, לדוגמה יצירת השהייה של בין 10-30 שניות בין ניסיון לניסיון בצורה רנדומלית.
- מניעת חשיפת שגיאות במנגנון האימות המעידות על קיום או אי קיום משתמש.
- החזרת סטטוס קוד 200, בעת כשל בתהליך האימות כך שלא תהיה אינדיקציה על אימות תקין על סמך קוד הסטטוס החוזר מהשרת.

2.2.10 "פיצוח" סיסמאות

- יש להצפין סיסמאות בהצפנה חד כיוונית בעזרת אלגוריתם חזק (לכל הפחות SHA256).
- יש להשתמש ב-salt רנדומלי לכל סיסמה הנוצרת לצורך מניעת hash זהה ליותר ממשתמש אחד (unique salt).
- אין להשתמש ב-salt זהה לכל הסיסמאות.
- אין להשתמש ב-salt המורכב מפרטי המשתמש אלא ברנדולי.
- אין להשתמש ב-salt קצר מדי היות והדבר יקל על התוקף, ההמלצה הינה שימוש ב- salt שזהה לגודל ה- hash הנוצר ביצירת הסיסמה,

לדוגמה בהצפנת SHA256 הפלט הינו 32 bytes, לכן מומלץ ליצור salt רנדומלי באותו הגודל.

- הצפנה כפולה יוצרת בעיות ואינה מהווה פתרון יותר מאובטח, על כן אין להשתמש בשיטה זה.
- אין להשתמש בשיטות הצפנה שאינן מאובטחות כגון md5, base64, sha1.
- Salt צריך להיווצר בצורה רנדומלית באמצעות מנגנון המיועד לכך כגון: Cryptographically Secure Pseudo-Random Number Generator (CSPRNG), ולא בשיטות קוד פשוטות כגון rand(). דוגמה: https://msdn.microsoft.com/en-us/library/system.security.cryptography.rngcryptoserviceprovider(v=vs.110).aspx
- אין להשתמש בשיטות הצפנה שאינן מאובטחות כגון md5, base64, sha1.
- שמירת ה-SALT תישמר יחד עם ה- HASH בבסיס הנתונים תחת רשומת המשתמש.
- אין להשתמש בהצפנה או חשיפת ה- HASH בקוד צד לקוח.

דוגמה נכונה ליישום ב#C:

https://crackstation.net/hashing-security.htm#aspsourcecode

2.2.11 תהליך החלפת\איפוס סיסמה

- יש לוודא כי בעת תהליך החלפת סיסמה חובה על המשתמש להזין את סיסמתו הנוכחית.
- יש לוודא כי בכל פעולה של עדכון פרטים מזהים של החשבון, קיים מנגנון הגנה מפני התקפות CSRF, כגון שימוש ב-CSRF Token.
- יש לוודא כי בתהליך איפוס סיסמה, פרטי הזיהוי של מבקש הבקשה נבדקים ולא מאפשרים לתוקף לבצע מניפולציות בבקשה הנשלחת לשרת במטרה לאפס סיסמה למשתמשים.
- יש לוודא כי בתהליך איפוס סיסמה, אין שימוש בהזנת פרטים ו\או שאלות אבטחה הקלות לניחוש או לחלופין נעשה שימוש בפרטים שקל לקבלם כגון ת.ז ותאריך לידה.
- יש לוודא כי בעת שימוש בשאלות אבטחה, יש לשמור על סודיות השאלות והתשובות לאורך כל התהליך באמצעות הצפנת הנתונים, בנוסף יש לשמור על הכללים הבאים:
  - הצגת השאלות בצורה רנדומלית אשר לא תאפשר לדעת מראש מה תהיה השאלה הנשאלת.
  - יש לוודא כי בעת הצגה של יותר משאלה אחת, המשתמש אינו מזין תשובה זהה (כמובן יש להתייחס לכך בעת יצירת השאלות).
- יש להגביל את כמות הבקשות השגויות לשינוי סיסמה באותו מנגנון המשמש את אימות המשתמש למערכת.
- יש להגביל את כמות הפעמים בהן משתמש מנסה לשחזר את סיסמתו לאחר ניסיונות מרובים במן סביר.
- יש להטמיע CAPTCHA בתהליך שחזור\איפוס סיסמה.

2.2.12 יש להגביל גישה לבסיס הנתונים המחזיק את סיסמאות המשתמשים ולוודא שאין גישת לשינוי או כתיבה לגורמים ואפליקציות שלא לצורך בכדי למנוע מגורם זדוני המקבל גישה למערכת לשנות סיסמאות של משתמשים וכו'. כמו כן אין לספק הרשאת קריאה ללא צורך.