ממשל זמין – פרויקט תהיל"ה

הטמעה של אפליקציות WEB

בפרויקט תהיל"ה

גרסא 9

ממשל זמין – פרויקט תהיל"ה

מאפייני מסמך

מחברים	מאיר קראוסהר, נימרוד לוריא
מספר גירסה	9
סטטוס	הפצה
תאריך הוצאה	נובמבר 2010
שם קובץ אלקטרוני	NewSiteProc

תשומות / הערות

שם/תפקיד	הערה (אופציונאלי)	תאריך	חתימה

אישורים

שם/תפקיד	תאריך	חתימה

היסטוריה

מ. גרסה	ת. הוצאה	מחבר	שינויים מרכזיים בגרסה

הפצה

מ. גרסה	נמענים

ממשל זמין – פרויקט תהיל"ה

תוכן עניינים

1. כללי

פרויקט תהיל"ה מספק שירות אירוח של אתרי אינטרנט עבור משרדי הממשלה.

עיקר פעולתם של האתרים הממשלתיים מתבטא במתן מענה לשלושה צרכים עיקריים:

- שירותים מקוונים במסגרת ממשל זמין (תשלומים, טפסים, פניות ציבור וכו')
- מקור מידע רשמי ועדכני של נתונים, פרסומים והודעות לכלל הציבור
- חלק ממערך ההסברה של מדינת ישראל

המידע המאוחסן על שרתי האינטרנט בתהיל"ה, הינו רשמי ובדרך כלל רגיש.

מערכות אלה נתונות תחת נסיונות התקפה קבועים. נסיונות השחתה, החדרת תעמולה, גניבה, שיבוש מידע, השבתה ומניעת שירות.

חולשת אבטחה אחת יכולה להספיק בכדי להשתלט על מערכת, ולהשתמש בה כעוגן להמשך התקפות לתוך רשת האירוח.

ללא קשר לחשיבות המידע אותו הם מציגים, כל אתרי האינטרנט המתארחים בפרויקט נתונים תחת ניסיונות פריצה והתקפה, ולכן שירות זה כפוף למדיניות אבטחת מידע נוקשה.

כלל המפתח ביישום מדיניות זו הוא, שכל פגיעה באתר או מערכת ממשלתיים, כמוה כפגיעה בנכס ממשלתי ולצורך העניין, פגיעה בממשלה. עקרונות אבטחת המידע נקבעו ואושרו ע"י החשב הכללי, מנהלת הרשת הממשלתית, אגף ביטחון באוצר וחברות אבטחת מידע.

מסמך זה נותן כללי מסגרת ודרישות מינימום להכנסת אפליקציות חיצוניות על מערכות בפרויקט תהיל"ה.

1.1 תיאור האיומים

שלוש רמות עיקריות:

- **הרשת (Network)** (שימוש בפרוטוקולים אסורים, הצלבת רשתות,...)
- **מערכת ההפעלה** (הרשאות לקויות, שימוש בשירותים אסורים,...)
- **היישום (Application)** (עיבוד שגוי של קלט משתמש, הרשאות, ...)

סכנות ואיומים נפוצים על יישומי אינטרנט:

- השחתה "רועשת" (Defacement) – השחתת תצוגה לצרכי תעמולה
- השחתה סמויה – הצגת מידע שגוי, זיוף נתונים (ובכך פגיעה באמינות המערכת, ובמהימנות של ממשל זמין ובעל המידע כנותני השירות)
- מניעת שירות – האטת המערכת או השבתתה
- גניבת זהויות – ביצוע פעולות תוך התחזות, ניצול הרשאות אחרות
- הונאות, הפעלת יישומים במירמה – גניבת כסף, גניבת מידע
- השתלטות על המערכת לצורך חדירה פנימה לתוך הארגון
- הונאת משתמשים – פישינג, ריגול אחר משתמשים

טכניקות פעולה:

- Reconnaissance – איסוף מידע על המערכת. מיפוי מרכיבים, מעה"פ, סביבת פיתוח, גרסאות, זיהוי תהליכים. מיפוי משתמשים, סריקת סיסמאות, כתובות, פורטים פתוחים, שירותים פעילים וכו'
  איסוף המידע נעשה בדרכים מגוונות. טכניקה נפוצה הינה שימוש שגוי בכוונה. ייצור מכוון של שגיאות במערכת מפיק ההודעות אשר חושפות מידע פנימי אם אינן מנוהלות כראוי. זה מתבצע ע"י כלי סריקה אוטומאטיים, ובצורה ידנית.
- באמצעות המידע שהתקבל ניתן לתקוף חולשות ברכיבים שנחשפו:
  - ניצול פרצות אבטחה ברכיבים מגרסאות ישנות ולא מעודכנות
  - ניצול לרעה של ממשקי ניהול שנתגלו
  - זיהוי ממשקי העלאת קבצים ונסיון להעביר וירוסים, רוגלות וכו'
  - הרצת קוד, למשל באמצעות זליגת חוצצים (Buffer Overflow)
  - טיפול לקוי בקלט משתמש: הזרקת קוד, שיבוש המידע שמועבר למערכת:
    - Cross-site scripting
    - injections: HTML, XML, LDAP, SQL
  - ניצול הרשאות לקויות
  - ניצול מנגנוני הזדהות חלשים, ניצול מנגנון ניהול משתמשים לא מאובטח
  - הקלטת תעבורה לא מוצפנת Man in the Middle
  - וכו'...

אכיפת מדיניות אבטחת המידע מתבצעת כחלק מהפעילות השוטפת.

תנאי הכרחי ביישום המדיניות הינו הקפדה על ארכיטקטורה ותכנון נכון של תהליכים, פיתוח קוד והקשחת מערכת בהתאם לתקני האבטחה.

2. הקמת אתר בתהילה - נהלי עבודה ודרישות

2.1 תהליך הקמת אתר

<u>מוקדם ככל שניתן: פגישת התנעה</u>

- הצגת המערכת
- תיאום ציפיות. הסכמה על דרישות, פונקצינאליות, לו"ז
- קביעת אנשי קשר: מטעם בעל האתר, חברת הפיתוח, מנהל פרויקט של ממשל זמין
- קביעת ארכיטקטורה מוסכמת

בהמשך לפי הצורך: פגישות סינכרון ותיאום ציפיות (SLA, בעיות טכניות וכו')

<u>לפחות</u> חודש לפני מועד העלייה לאוויר המתוכנן – תחילת פעילות בתהילה:

- **התקנת מערכת הפעלה, ועדכוני אבטחה** – ע"י צוות תהילה
- **התקנת היישום** – לפי תיאום מוקדם ביחד עם נציג צוות תהילה

  בסיום:
  - הפעלת המערכת ובדיקת תקינות
  - גיבוי חיצוני מלא
- **הקשחה** – הטמעת תבניות והגדרות לפי נוהלי תהילה

  בסיום:
  - בדיקות תקינות והתאמות להקשחות
  - גיבוי מלא נוסף של המערכת

- **ביצוע בדיקת קבלה** – באחריות אחראי האתר (ורק לאחר הקשחה)
- **אישור תצורה סופית**
- **בדיקת אבטחה** (Penetration Test) – תבוצע לאחר שהמערכת *<u>מותקנת בתצורתה הסופית ברשת היצור בתהילה</u>*. שינויי גרסא/קוד וכו' יצריכו ביצוע הבדיקה שנית. הבדיקה תבוצע ע"י גורם שלישי (שאינו נמנה על צוות הפיתוח או ההקמה). הגוף המבצע חייב באישור של צוות תהילה. מצורף נספח
- **תיקון ליקויים** לפי ממצאי הבדיקה והדוח שפורסם (פעילות זו עלולה לדרוש זמן. יש להיערך לכך עוד בשלב קביעת הלו"ז)
- **בדיקת תיקון ליקויים** – לפי ממצאי הדוח שפורסם
- **חשיפה** – פירסום DNS, הפעלת application FW, פתיחת כתובת ה –IP

2.2 דגשים:

- **צוות תהילה מלווה את הפרויקט עוד משלב האפיון, ולאורך כל התהליך.**
- **מנהל הפרויקט שהוקצה ע"י ממשל זמין הוא איש הקשר.**
- **לפני מימוש ארכיטקטורה יש לוודא אישור צוות תהילה על התצורה המוצעת**
- **התקנות:**
  - **מעה"פ והקשחות: *ע"י צוות תהילה* בחווה בתהילה.**
  - **התקנת היישום: ע"י הלקוח, ביחד עם צוות תהילה במועד קבוע מראש.**
- **המערכת חייבת להיות מסוגלת לפעול תחת ההגבלות וההקשחות של תהילה.**
- **המערכת לא תיחשף לפני סיום בדיקת אבטחה!**
  - **בדיקת האבטחה תבוצע ע"י גורם חיצוני ובלתי תלוי, *מאושר ע"י תהילה.***
  - **באחריות הגוף הבודק לספק דו"ח מפורט: מה נבחן, ומה הממצאים.**
  - **באחריות הלקוח לתקן.**
- **לא תתאפשר השתלטות על מערכת ההפעלה מרחוק (RDP, SSH וכו')**
- **לא תתאפשר עריכת תוכן, ניהול משתמשים ברשת האינטרנט:**
  - **אלא אם זיהוי ואימות הישויות מתבצע בתשתית ה- PKI הממשלתית (תמו"ז).**
  - **פורומים, תגובות וכו' של הציבור, *מחייבים אפיון מדויק ומודרציה*.**
  - **לא תתאפשר כל צורה של ניהול הרשאות.**
- **לא תתאפשר העלאת קבצים מהאינטרנט ללא סריקה וזיהוי.**
- **גולשים מוגדרים כמשתמש אנונימי ומאופשרים לקריאה בלבד.**
- **יש לוודא שלא מתפרסם באתר שום מידע פרט למה שתוכנן. אסור לחשוף שום מידע פנימי דוגמת כתובות IP, שמות שרתים, שמות משתמשים, גרסאות תוכנה וכו'. יש לוודא שהמערכת *תמיד* מציגה הודעות שגיאה ערוכות, ללא חשיפת שום מידע.**
- **במערכות ניהול תוכן (מוצרי content management למיניהם):**
  - **המערכת תורכב משרת "קדמי" R/O ושרת "אחורי" R/W.**
  - **עריכת תכנים רק על השרת האחורי (התצורה נקבעת עם צוות תהילה).**
- **עבודה מול בסיס נתונים:**
  - **Connection String מוצפן.**
  - **סיסמאות ונתונים חסויים חייבים להיות מוצפנים.**
  - **לעולם לא גישה ע"י אדמיניסטראטור.**
  - **לעולם לא באמצעות שאילתות דינאמיות יש להשתמש בפרוצדורות.**
  - **גישה עם משתמש בעל הרשאות קריאה בלבד. הרשאת כתיבה או מחיקה. תוגדר נקודתית על המשאב הרלוונטי. אין הרשאות גורפות.**

- **הפרדת תחומי פעילות: בסיס נתונים, שרת DC וכו' על שרתים נפרדים.**

על כל חריגה מדגשים אלה יש לדווח לצוות תהילה

אי עמידה בתנאים אלה עלולה לגרור הפסקת פעילות גם במצב ייצור

2.3 נקודות נוספות:

- בכל גישה של גולשים לאזורים המחייבים זיהוי ואימות משתמשים יש להצפין את ה-session (שימוש ב- SSL)
- **התקנת תוכנות:** לשרת לא יינתנו שום יכולות פרט לאירוח אתרים. עם זאת, התקנה של אפליקציות נוספות תתאפשר, אם יתברר שהן הכרחיות לפעילות (ואם הן אינן בסתירה למדיניות תהילה). תוכנות client side דוגמת Office או סביבות פיתוח למיניהן (למשל Visual studio) אינן עונות על הדרישה הנ"ל ולא יותקנו על שרת.
  - התקנה של תוכנת צד ג' מצריכה אישור של צוות אבטחת מידע תהילה, בדבר אמינות היצרן, ומחייבת מעקב אחר שינויים ועדכוני גרסא.
- **שירותים נלווים:** כגון Media, DC ,DB וכו', ניתנים על שרתים נפרדים.
- **התקנה מינימאלית:** יש לוודא שאין שאריות פיתוח, קבצים וכל סוג של מידע לא רלוונטים, תיעוד חשוף וכיוצא באלה.
- **הצפנת נתונים חסויים בקובצי קונפיגורציה (או כל קובץ אחר) ובסיס הנתונים.**
- **התקנת המערכת על כונן נפרד ממערכת ההפעלה** (למשל **D:**).
- לוגים, מקורות מידע וכד' לא יוכנסו ל- root של האתר.
- **לא ניתן ליזום התקשרות אל מחוץ לרשת**. שרת האינטרנט עונה לפניות ולא יוזם אותן. חריגים: 1) שליחת אי-מייל או SMS מהשרת ניתן באמצעות שרת SMTP נפרד, יש לאפיין עם צוות תהילה 2) צריכת Web-Service חיצוני באמצעות ה- XML FW ("צומת השירותים הממשלתי" – קיים נוהל נפרד).
- **עמימות:** הקפדה על אי חשיפה של מידע מערכתי פנימי כגון כתובות, שמות משתמש וכו'. ביטול הצגת הודעות שגיאה מפורטות, הערות, וכל מידע לא רלוונטי למצב ייצור.
- **טיפול בשגיאות**: הגדרת מנגנון לכידת שגיאות. הצגת הודעה כללית במקרה תקלה
- **תיעוד**

- **בקרת גישה:** גולש חיצוני:
  - בעל הרשאות קריאה + "deny write"
  - מוגדר רק על מחיצות רלוונטיות
  - במקרה שיש צורך לכתוב נתוני שהוזנו ע"י גולש יש לאפיין עם צוות תהילה
- **הגדרת Firewall מקומי**
- **ניוון ה- Web Server:**
  - אתרים ואפליקציות **לא** על כונן C:\
  - איסור על נווט בין משאבים ע"י שימוש ב- ..\..\..
  - ביטול הודעות שגיאה
  - ניטרול יכולות אסורות (כגון WebDav), מחיקת ספריות מיותרות (למשל IISExamples)
  - אין אפשרות ניהול מרחוק
  - הגדרה כ- read only, ביטול directory listing
  - שימוש במתודות GET ,POST,HEAD בלבד
  - extensions מותרים בלבד (html, asp, aspx, php, וכו')
  - הפעלת תיעוד (logging) מלא

2.4 שרתים נוספים

שרת אינטרנט חשוף לבקשות חיצוניות, ועל כן לא ניתנת לו שום פונקציונאליות פרט לאירוח אתר.

לצרכים אחרים מוקצה שרת ייעודי:

- עדכוני תוכן – שרת Read/Write אחורי
- שרת DB
- שרת DC
- שרת BPM (BizTalk)
- שרת Media
- וכו'...

שרתים אלה מופרדים גם לצורך ייעול תהליכים והפחתת סיבוכיות תפעולית. ניתן להשתמש בשרתים שיתופיים של תהילה.

כל השרתים עוברים אותה סדרת הקשחות ופועלים תחת אותם נהלי עבודה שהוזכרו בסעיף 2.1.

2.5 תקשורת

שרתי האינטרנט מחוברים באמצעות שני כרטיסי רשת:

- **<u>כרטיס רשת "קדמי"</u>** – מקבל בקשות גולשים מהתווך הציבורי
  - נגיש בפורט 80 ו- 443 בלבד
  - FW מקומי מופעל ומפלטר בנוסף להקשחות על הכרטיס. מקבל הגדרות מתבניות הקשחה

- **<u>כרטיס רשת "אחורי"</u>** – פעולות אדמיניסטרטיביות:
  - פעילות בפורטים סטנדרטיים. תצורות אחרות מחייבות אישור והקשחות נוספות.
  - משמש ל:
    - עדכוני תוכן מרחוק
    - הכנסת, שליפת מידע
    - תקשורת פנימית, מול ה- DB וכו'
    - צריכה או פרסום WS דרך "צומת השירותים הממשלתי"
    - גיבויים
    - עדכוני אבטחה

<u>פעולות System לא יתאפשרו מרחוק (התקנות, הפעלת service,רישום DLL וכו')</u>

FW מקומי מופעל כנ"ל

- **במקרה הצורך כרטיסים נוספים**:
  - תעבורה פנים דומיינית
  - Heartbeat
  - כל הגדרה שתדרש

- שרת אינטרנט קדמי היא היחיד שחשוף לאינטרנט באמצעות IP אמיתי. שרתים שאינם שרתי אינטרנט קדמיים יש רשת פנימית בלבד.
- שימוש בפורטים לא סטנדרטיים מחייב אישור FW וסגמנטציה ברשת.
- **לא יתאפשר ייזום תקשורת מול רכיבים מחוץ לרשת תהילה.**

כלל זה לא חל על צריכת שירותי רשת חיצוניים (Web Services) אלא אם הוגדרו באמצעות צומת השירותים הממשלתי ועפ"י תקן WS.gov.il. יש לאפיין עם צוות תהילה ולתאם אינטגרציה בהתאם.

2.6 משתמשים, הרשאות, מדיניות סיסמאות ו- auditing

2.61 ***<u>תצורת משתמשים:</u>***

- **הצפנה:** בכל גישה של גולשים לאזורים המחייבים זיהוי ואימות משתמשים יש להצפין את ה- session (שימוש ב- SSL)
- **נתוני משתמשים בבסיס נתונים או קבצי קונפיגורציה יוצפנו**
- סיווג הפעילות של כל משתמש על המשאבים : קריאה, כתיבה, או הרצה לא ינתנו הרשאות "Full Control"
- גישה לבסיס נתונים: באמצעות משתמש עם **הרשאות קריאה בלבד**.
אם יש צורך בהרשאות כתיבה, יש להגדיר נקודתית. בהקשחת בסיס הנתונים נמנעת גישה לטבלאות ופרוצדורות מערכת.
- ניטרול guest, שינוי שמות דיפולטיים (administrator, sa)
- גישה למשאבי מערכת ההפעלה ניתנת רק לקבוצת אדמיניסטרציה (ורק מקומית על השרת)
- סיווג משתמשים לפי ארבע קטגוריות:
  - קבוצת ניהול (אדמיניסטראטורים, סיסטם)
  - מעדכני תוכן: שליטה מלאה על הנתונים שלהם
  - הרצת תהליכים: משתמשים מובנים להפעלת שירותים
  - User/Internet user: הרשאות קריאה
- אין קיבוץ משתמשים גורף – "everyone" לצורך מתן הרשאות

2.62 ***<u>מדיניות משתמשים וסיסמאות:</u>***

- **מדיניות סיסמא:**
  - הסיסמא תהיה בת 8 תווים לפחות
  - הסיסמא תכיל ספרות, לפחות 2 אותיות גדולות, ו- 2 אותיות קטנות
  - הסיסמא תכיל תו אחד לפחות שאינו אלפא נומרי (^%$#@!)
  - *<u>הסיסמא לא תהיה זהה לשם המשתמש</u>*.

- **מנגנוני Anti-Brute Force**
  - מניעת הצפה של קלט משתמש: יש להשתמש במנגנון Captcha לפי מפרט שמצורף בנספח.
  - משתמש לא קיים: סגירת ה – session לאחר 6 נסיונות כושלים
  - חידוש סיסמא יבוצע באמצעות פרטים מזהים נוספים (דוגמת כתובת e-mail ואלידית, שאלות מנחות וכד'). **חובה לאשר עם צוות תהילה**.
  - בהגדרת משתמש חדש: יש לבצע ולידציה על כתובת המייל מול הUser

- **מניעת ניחוש משתמשים, ונעילת משתמשי המערכת**
  - *<u>בכל נסיון זיהוי</u>* שם משתמש וסיסמא יש להפעיל מנגנון Captcha, כך שלא ניתן יהיה להפעיל סקריפט לניחוש סיסמאות.
  - <u>מניעת DOS</u> – בכדי למנוע נעילה מכוונת של משתמשי מערכת, אם בתהליך ההזדהות טעה מספר רב של פעמים בהקשת הסיסמא אין לנעול את המשתמש אם שם המשתמש אכן קיים במערכת.
    ללא תלות בקיומו של המשתמש כן או לא, לאחר מספר נסיונות כושלים יש לסגור את ה- session, ולוודא ש- captcha מופעל בכל נסיון זיהוי כאמור לעיל.
  - *<u>אין להציג הודעות שגיאה כגון "שם משתמש לא קיים", "סיסמא שגויה"</u>* וכו'
    - **יש להציג הודעת שגיאה כללית**
    - יש לוודא שב- html לא חוזר error דוגמת "login error"

- **התיישנות סיסמאות**
  - יש לבצע החלפת סיסמאות תקופתית (לפחות פעם בחודשיים באפליקציה רגישה), סיסמאות יחויבו להיות מורכבות לפי מדיניות תהיל"ה.
- **היסטוריית סיסמאות**
  - יש לבדוק היסטוריית סיסמאות (לפחות 5 סיסמאות אחורה באפליקציות רגישות) ולמנוע ממשתמש למחזר סיסמאות.

- **הודעות מפורטות**
  - אין לתת הודעות מפורטות (כגון "שם משתמש לא נכון / סיסמא לא נכונה") אלא להשתמש בהודעה גנרית דוגמת "חלה שגיאה אנא פנה לתמיכה".

- **נתוני זיהוי נוספים**
  - במידה והאפליקציה רגישה, יש להוסיף עוד נתון זיהוי כגון מספר עובד / ת.ז.

- **הגדרת תהליך חידוש סיסמא:**
  - גלישה לאיזור חידוש סיסמא
  - הכנסת פרטים מזהים כפי שהוגדר מראש, דוגמת דואר אלקטרוני, ת.ז. ופרט מזהה נוסף (מספר עובד, מספר טלפון, שם בית ספר, וכו').
  - שליחה של מייל מהשרת ללקוח עם URL חד-פעמי (Random URL GUID/Generator) שמפנה לדף חידוש סיסמא.
  - URL זה אקטואלי למשך שעה

<u>שימוש בכרטיסים חכמים</u>

מענה לנקודות הנ"ל, ושיפור משמעותי של אבטחת זיהוי המשתמשים מתקבל ע"י שימוש בכרטיס חכם של ממשל זמין. יש לאפיין עם צוות תהילה.

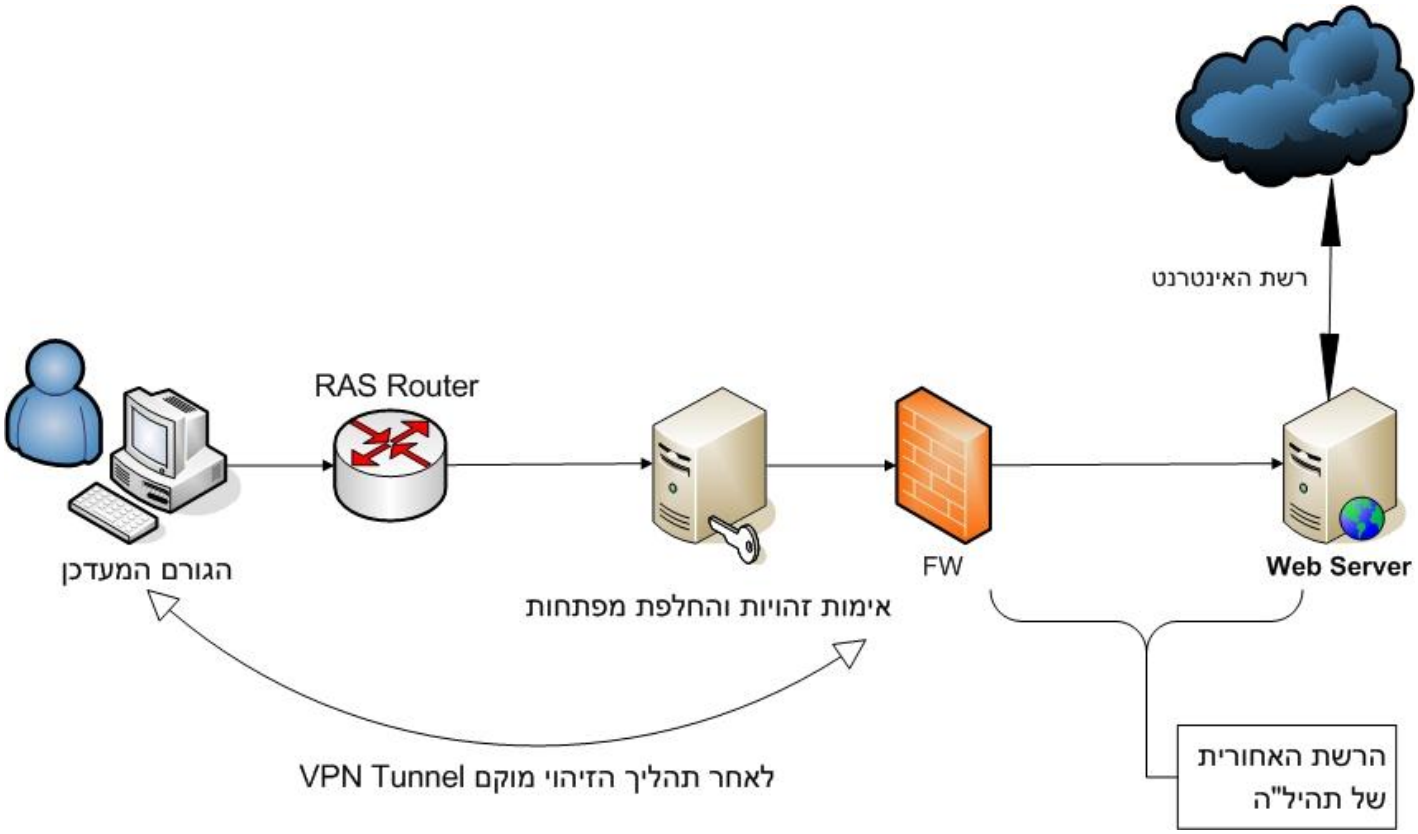
2.7 עדכון תוכן

ניצול ממשק לעריכת תוכן של אתר הוא השיטה השכיחה ביותר להשחתת אתרים או השתלטות על מערכות. כיוון שזהו ממשק "אטרקטיבי" במיוחד, המדיניות הנהוגה בפרויקט תהיל"ה בכל הקשור לנושא זה הינה מחמירה ביותר

- ***התצורה שבה בעל המידע מנהל את האתר ועורך בו תוכן חייבת איפיון ואישור ע"י צוות תהילה בלבד.***
- **פעילות זו נבחנת באופן קבוע.**
- **סטייה מהתצורה שאושרה עלולה לגרור הפסקת הפעילות ואף השבתת האתר. כל פתרון שלא הובא לידיעת הצוות, לא יתאפשר.**

1. חל איסור מוחלט לבצע דרך תווך ציבורי (רשת האינטרנט):
   1.1. כל סוג של העלאת קבצים
   1.2. כל סוג של נסיון לנהל את האפליקציה
   1.3. כל סוג של נסיון לנהל את מערכת ההפעלה
   1.4. כל סוג של ניהול משתמשים
2. לא תינתן יכולת להשתלט על השרת (VNC, RDP, SSH, וכו')
3. עדכוני תוכן ניתן לבצע רק דרך הרשת "האחורית" (רשת האדמיניסטרציה). התחברות לרשת זו לפי הנחיות צוות תהילה
4. **אי עמידה בתנאים אלה עלולה לגרור הפסקת פעילות**

ארכיטקטורת עדכון תוכן:



<u>מאפייני ההתקשרות:</u>

- ממשק עריכת הנתונים לא יכול להיות בשום תצורת client server פרט לדפדפן.
- גישה לרשת העדכון מתבצעת ע"י התחברות דרך קו ייעודי, בד"כ ADSL או ענן IP-VPN אולם כל דרך אחרת יכולה להיות נדונה.
- ***התחברות לשרת אינטרנט ברשת האחורית של תהילה תבוצע מתחנת עבודה ייעודית, אשר אינה מחוברת לשום רשת אחרת (stand alone).*** במילים אחרות, *<u>עמדת העדכון לא תחובר לא לרשת המשרדית, ולא לאינטרנט</u>*.
  יחד עם האמור לעיל ניתן לקשר עמדת עדכון לרשתות אחרות אם סנכרון המידע מתבצע ב- WS דרך צומת השירותים, או בשימוש בכרטיס חכם, או במנגנון לגישור בין רשתות מופרדות (כגון מנגנון כספות, מחשב בודל וכו'). יש לאפיין ולאשר עם צוות תהילה.
- מצפנות (S-box, PIX), סרטיפיקטים יסופקו ע"י תהיל"ה בתשלום.
- מאופשרת פעולה בפורטים ספציפיים. כל בקשה חריגה יש לציין מראש .
- יצירת הקשר היא תמיד אל הרשת. לא ניתן ליזום התקשרות מהשרת החוצה
- ניתן לקבל נתונים מהשרת ב-SMTP או SMS או WS.
- ניתן לתשאל מידע מרוחק באמצעות Web Services לפי תקן WS-gov.il בלבד.

צורות עדכון תוכן:

- העברת קבצים ב- FTP
- ממשק עריכה דרך HTTP בלבד
- Web Services לפי תקן WS-gov.il

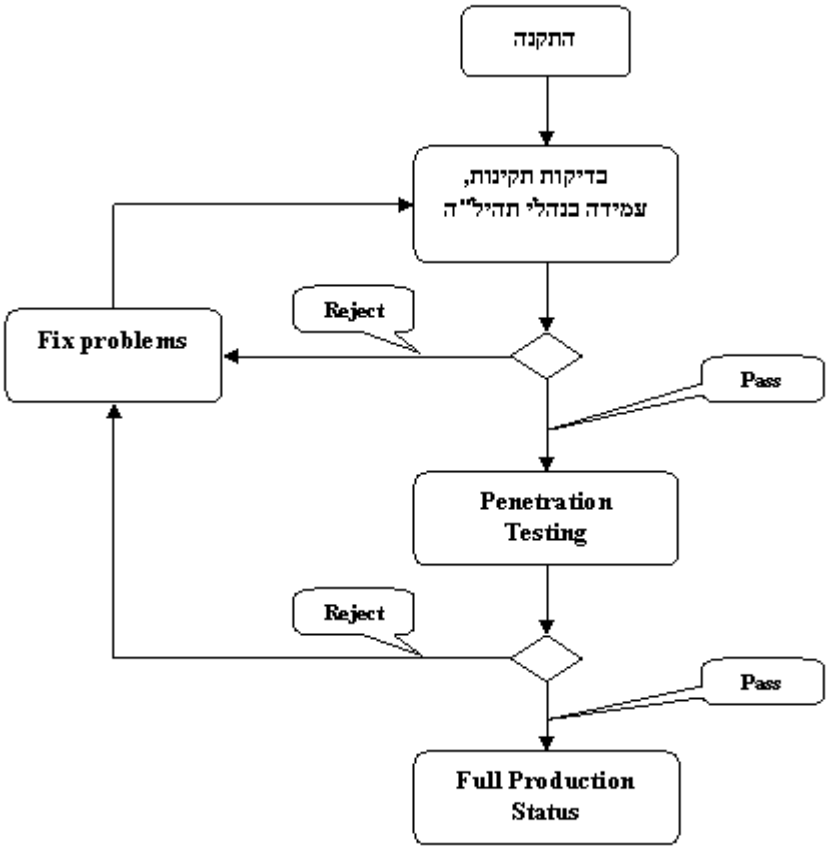
2.8 שימוש בשירותים קיימים בתהילה

- שירות תשלומים
- שירות טפסים
- שירות מכרזים
- שירותי PKI חתימה, זיהוי והצפנה
- שירותי רשת (רק עפ"י ws.gov.il)
- שירות מדיה
- שירות Caching

3.הנחיות לפיתוח מאובטח

מתכנתים רבים, מנוסים ככל שיהיו בפיתוח אפליקציות, לעיתים אינם מודעים לסכנות ולבעיות שעלולות להיווצר בעת שהמערכות שלהם עוברות לסביבת אינטרנט.

כל אפליקציית web שמועמדת לעבור למצב production בחסות תהיל"ה, עוברת סדרה של בדיקות שנועדו להבטיח עמידה בסטנדרטים של ביצועים ואבטחת מידע.



מסלול אישור האפליקציה המתואר לעיל הינה תנאי הכרחי לפני מעבר למצב ייצור.

יש לתת תשומת לב מיוחדת לנושאי אבטחת מידע במשך כל מחזור החיים של האפליקציה ובעיקר כשמתעורר הצורך לבצע שינויים מבניים, שינויי לוגיקה, זרימת מידע, ושינויי תצורה. העקרונות האבטחתיים הפרושים במסמך זה תקפים ומחייבים הן לשלב העלאת אפליקציות חדשות והן לגבי עדכונים ושינויים באפליקציות קיימות.

<u>רשימת ה"לאווים" לפיתוח יישומים בסביבת האינטרנט:</u>

- לעולם לא לסמוך על מידע שמגיע מהמשתמשים (כולל נתונים שלא אמורים להיות מוזנים על ידם – למשל cookies)
- לא להניח דבר על מידע שמגיע ממשתמש
- לעולם לא להמעיט בכוונות הזדוניות של משתמשים
- לא לספק שום מידע פרט לזה שצריך להתפרסם
- לא להחצין/להציג שגיאות
- להימנע מ- client side logic בלבד
- לעולם לא הרשאות גורפות, תמיד לתת הרשאות מינימאליות
- לעולם לא לגשת לבסיס הנתונים עם הרשאות אדמיניסטרטור

3.1 מדיניות סיסמאות

- הסיסמא לא תעבור גלויה ברשת אלא בצורה מוצפנת \ ב hash או על גבי תווך מוצפן.
- המערכת תספק למשתמש את היכולת להחליף את הסיסמא בעצמו, בצורה בטוחה, בכל עת.
- אורכם של שמות המשתמשים של המערכת יהיה לפחות 6 תווים.
- לא יוגדרו במערכת משתמשים בעלי שם משתמש טריוויאלי, כגון 'admin'.
- לא יתאפשר שם משתמש וסיסמא זהים
- במערכות רגישות, סיסמת המשתמש לא תהיה קצרה מ-10 תווים ותהיה מורכבת מ-4 קבוצות תווים הבאות:
  - אותיות קטנות
  - אותיות גדולות
  - ספרות
  - תווים מיוחדים
- סיסמתו של מנהל המערכת (אדמיניסטראטור) תהיה באורך של 12 תווים לפחות.
- תלוי ברגישות המערכת, תוקפה של סיסמת משתמש יפוג החל מ- 60 יום ועל המשתמש יהיה להחליף את סיסמתו בהתאם למבנה המתואר לעיל.
- תלוי ברגישות המערכת, מנגנון החלפת הסיסמא ישמור היסטורית סיסמאות של 5 מחזורים לפחות, ולא יאפשר למשתמש לחזור על אף אחת מהסיסמאות הללו בעת החלפת הסיסמא.
- טרם החלפת הסיסמא על המשתמש יהיה להקיש את סיסמתו הנוכחית.
- החלפת הסיסמא לא תתאפשר בטווח של 24 שעות מהחלפת הסיסמא האחרונה.
- הסיסמא הראשונית של המשתמש תהיה רנדומאלית ובהתאם למבנה שהוגדר לעיל.
- המערכת תחייב את המשתמש להחליף את סיסמתו הראשונית בעת ההתחברות הראשונה למערכת.
- תוקף הסיסמא הראשונית יהיה 3 ימים, ולאחר מכן המשתמש ינעל ולא יוכל להשתמש בה.
- במקרה בו המשתמש שכח את סיסמתו, המערכת תיצור לו סיסמא חדשה. כמו הסיסמא הראשונית, סיסמא זו תהייה מוגבלת בתוקף והמשתמש יהיה מחויב להחליפה בעת השימוש הראשון בה.

- יצירת סיסמא חדשה במקרה בו המשתמש שכח את הנוכחית תהייה אך ורק לאחר זיהוי המשתמש באמצעים אחרים, כגון כתובת דואר אלקטרוני, שאלות סודיות וכדומה.
- אין להציג בשום שלב במחשבי המערכת, במחשבים של משתמשי המערכת, בקוד המקור של דפים וטפסים המועברים למשתמש, את מזהי האימות של המשתמשים השונים במערכת.
- סיסמת המשתמש תשמר בצורת hash בבסיס המידע.

3.2 נעילת משתמשים

- יש לבחון נעילת משתמשים לאחר מס' ניסיונות הזדהות כושלים ע"מ למנוע השבתת שירות . יש ליישם מנגנוני מניעת הצפה, כאמור בסעיף 2.62 מניעת Brute force
- בנעילת משתמש עדיף לא להשתמש במנגנוני שחרור אוטומטי, אלא השחרור יבוצע על ידי מנהל המערכת לאחר קבלת הפניה מהמשתמש ווידוי זהותו.
- נעילת המשתמש תתבצע בצד שרת המערכת ולא ברמת ה-Session או ה-Client.
- בכל מקרה משתמש ניהול המערכת לא ינעל לאחר ניסיונות זיהוי כושלים על מנת למנוע מצב של מניעת שירות של המערכת.

3.3 ניהול משתמשים והרשאות

- הרשאותיהם של המשתמשים יקבעו לפי עקרון ההרשאות המינימאליות הדרושות, כלומר כל משתמש מערכת יקבל את הרשאותיו בהתאם לדרישות עבודתו במערכת ולא מעבר לכך.
- ***<u>ביצוע פעולות ב- DB עם משתמש בעל הרשאות מינימאליות. לעולם לא עם משתמש אדמיניסטרטיבי</u>***
- הרשאות המשתמש ייבדקו בכל השכבות ובכל הרכיבים של המערכת.
- לעיתים מתחייבת אותנטיקציה נוספת בחלקים קריטיים של האפליקציה.
- ***יש לוודא שהמשתמשים הם רק אלה לסביבת היצור***. שאין שום משתמשי טסט, פיתוח, או שאריות אחרות למינהן temp, test, demo, debug וכו'..

- יש לבצע בדיקת הרשאות משתמש בכניסה לכל דף במערכת.
- יש לבצע בדיקת הרשאות משתמש טרם ביצוע פעולות במערכת.
- אין להסתמך על מנגנון זיהוי כמנגנון הרשאות. משתמש מזוהה במערכת אינו בהכרח מורשה לכל חלקיה.

בקרת הגישה תתבצע בצד השרת בלבד ולא תסתמך על נתונים השמורים במחשבו של הלקוח, לדוגמא cookies.

3.4 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. ולכן אסור להניח דבר על קלט פרמטרי.

<u>תמיד יש לאמת קלט!!</u>

- בדיקות ואימות הקלט יתבצעו גם בצד השרת וגם בצד הלקוח ויכללו בין היתר: בדיקת אורך הקלט, ווידוא שיש רק תווים מותרים (<u>white list</u>).
- יש להימנע ממתן יכולת "free format input" ולהגדיר הגבלות ככל שניתן, למשל בחירה מתפריט לעומת תיבת טקסט.

☒ ***<u>לא להסתמך רק על בדיקות בצד לקוח!!</u>***

☒ ***<u>לא להשתמש בערכים שהתקבלו ישירות מהמשתמש לצורך יצירת דף דינאמי.</u>***

☒ ***<u>לעולם לא להשתמש בשאילתות דינאמיות בעת פניה לבסיס הנתונים עם קלט מהמשתמש!! יש להשתמש בשאילתות פרמטריות או stored procedures</u>***

- בקרת קלט מהמשתמש תיבדק בשכבות השונות בהתאם לסוג המידע שאמור להתקבל. שימוש ב regular expression – **white list**, קרי, סינון על פי ערכים מותרים ידועים מראש ולא שלילת ערכים. זאת משום שניתן להציג קלטים ביותר מצורה אחת על ידי שימוש בקידוד שונה.
- *<u>הבדיקות יתבצעו גם בצד המשתמש וגם בצד השרת!</u>*

- בגישה ל WS וגישות SOAP באמצעות XML, יש לבצע בדיקות לקלט שמועבר למערכת לפי סכמות XSD מוגדרות מראש לכל פעולה \ מתודה בשרות אליו מתבצעת הגישה.

3.5 הגנה על מידע רגיש

עבור אזורים חסויים, הצפנה של זרם המידע תוך שימוש ב- SSL אינה תמיד מספקת שכן מנגנון זה מספק הגנה ברמת התעבורה ואינו ספציפי לאפליקציה. נחוצה גם הצפנה ברמת האפליקציה להסוואת תוכן קריטי:

- יש להצפין נתונים רגישים במערכת (הן בקבצים והן בבסיס הנתונים).
  - קבצי קונפיגורציה – יש להצפין נתונים חסויים כגון שם משתמש וסיסמא, Connection String מול בסיס הנתונים וכו'.
  - בסיס הנתונים – כנ"ל, למשל עבור טבלת משתמשים וסיסמאות.
  - כל נתוני זיהוי של רכיבי תוכנה כלשהם, דוגמת נתוני הזיהוי של שרת האפליקציה לשרת בסיס הנתונים כאמור, וכו'.
  - כל מיקום של נתונים רגישים וחסויים של משתמשי המערכת.
  - במידה וקיימים קבצים (כגון קבצי Word, PDF, תמונות וכדומה) אשר מכילים מידע רגיש בבסיס הנתונים יש להצפינם גם כן.
  - מפתח ההצפנה יישמר במקום מאובטח על שרת המערכת, כגון ה-Registry. הגישה למפתח תוגבל לאפליקציה ולאדמיניסטראטור של השרת בלבד.
  - יש לבצע הצפנה של המפתח ע"י שימוש בהצפנת DPAPI, יש לשמור עותק של המפתח במקום מוגן נפרד (פיזי) למקרה שלא ניתן לשחזר את המפתח המקורי.

- אחסון סיסמאות באופן מאובטח יעשה באופן הבא:
  - הסיסמאות אינן דורשות הצפנה דו כיוונית כיוון שאין צורך באחזורם, לפיכך הסיסמאות ישמרו בבסיס הנתונים לאחר ביצוע HASH על ערכן.
  - לכל משתמש בעת יצירת סיסמא ייבחר ערך רנדומאלי אשר ישורשר לסיסמא טרם ביצוע ה-HASH. ערך זה נקרא ערך SALT.
  - ערך ה-SALT ישמר בבסיס הנתונים יחד עם פרטי המשתמש.
  - על מנת לבדוק כי הסיסמא שהמשתמש הזין הינה נכונה, משרשרים אליה את ערך ה-SALT מבסיס הנתונים ומבצעים על הערך החדש את פעולת ה-HASH שבוצעה בעת שמירת הסיסמא. אם ערך ה-HASH החדש תואם את ערך ה-HASH אשר שמור בבסיס הנתונים, הרי שהסיסמא נכונה.
- 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 למנוע את שמירת נתוני המערכת בספריית הקבצים הזמניים ובמנגנוני ה-Cache במחשב המשתמש.
- בחלקים קריטיים של אפליקציה רגישות יש לשקול אותנטיקציה פעם נוספת.
- יש מצבים (דוגמת מסך תשלום בכרטיס אשראי) שבהם יש להוסיף לאותנטיקציה של המשתמש, פרט לשם משתמש וסיסמא, גם נתונים נוספים כגון ת"ז, מס' עובד, שאלה סודית כלשהי וכו'..

3.6 הגנה על מידע בתעבורה

בעקבות רגישות המידע אין לאפשר העברת מידע בתווך אינטרנט ללא הצפנתו. מומלץ להוסיף הגבלות אלו ברמת האפליקציה.

כל התעבורה תתבצע בתווך מוצפן.

3.7 ניהול מופעי משתמשים (Session Management)

- יש להבטיח כי נתוני session נשמרים בצורה בטוחה במהלך חיי המערכת ובפעולת המערכת השונות המתבצעות עם האובייקטים \ משתמשים.
- יש להבטיח כי קיימת הפרדה בין ניהול הזהויות לבין שימוש ב session כך שלא יתכן מצב כי משתמש שלא ביצע הזדהות יוכל להשתמש ב session פעיל של משתמש שביצע הזדהות כנדרש(גניבת זהות), כלומר 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.
- *<u>אין לשמור מידע רגיש ב SESSION , במידה ונדרש יש לבצע הצפנה של מידע זה</u>*.
- 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).
- על המערכת להימנע במידת האפשר בשימוש ב- client side state management כגון view state, cookies, hidden files לצורך קבלת נתונים עבור session.
- 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.
- יש למנוע ביצוע גישות מרובות מאותו SESSION למערכת.

הנחיות נוספות לגבי ניהול מופעי משתמשים כפי המופיע במסמך האפיון (פרק 5) :

- 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. יש לקיים מנגנון Idle Timeout אשר יסיים את ה-Session של המשתמש לאחר מספר דקות מוגדר, כ-15 דקות, של חוסר פעילות במערכת.
- יש לקיים מנגנון Session Timeout אשר יסיים את ה-Session לאחר זמן ארוך של פעילות במערכת, כ-8 שעות. מנגנון זה נועד למנוע שימוש במערכת באמצעות סקריפטים וכדומה.

- יש לשקול את ניתוק Session במצבי שגיאה מסוימים. שגיאת security – במקרה שזוהתה שגיאת אבטחה באפליקציה, יש לסיים מייד את ה- session.
- ניתוק ה-Session יבוצע על ידי סיום תוקף ה-Session בצד השרת, ולא על ידי העברת הלקוח לדף הכניסה בלבד.
- Logoff/Logout – תמיד צריכה להיות למשתמש האפשרות לסיים את ה- session ולצאת בצורה מסודרת ובטוחה.

3.8 ניתוק מערכת

- האפליקציה תאפשר יציאה מסודרת ונוחה מהמערכת בכל דף החל מדף הכניסה (Login).
- ניתוק זה יבטיח כי משתמש לא יוכל לבצע שימוש חוזר במערכת ללא ביצוע הזדהות מלאה מחדש.
- במקרה של זיהוי פעילות חשודה במערכת (כפי שהוגדרה במידול הסיכונים) כגון ניסיונות לביצוע sql injection או הזנת סקריפטים זדוניים בשדות קלט, נדרש לבצע ניתוק כפוי של המשתמש, לבצע רישום ללוג וכן להתריע על כך למנהל המערכת.

3.9 שימוש בתעודות והצפנות

עבור מידע המוגדר כרגיש, יש לאפשר טיפול באמצעי מידור הן ברמת מנהלי המערכת והן ברמת המשתמש. כולל:

- תמיכה בסוגי מידע שונים.
- יכולת הגדרה במערכי ה-Audit לרישום גישה או ניסיונות גישה למידע המוגדר כרגיש. רישום ה-Audit יבוצע באופן מלא בכל שכבה, ובביצוע האחזור ניתן יהיה להפריד באופן מובהק בין התהליכים ובין השכבות השונות שבהם בוצע ה-Audit.

במערכת מידע נדרש לבצע הצפנה לפי הכללים הבאים:

כאשר מתבצעת הצפנה למידע רגיש יש לממש אלגוריתמי הצפנה לפי הכללים הבאים:

- אין לבצע שימוש באלגוריתמים שפותחו בצורה עצמאית.
- יש לבצע שימוש באלגוריתמים מוכרים כגון:
- AES עבור הצפנה סימטרית
- RSA עבור הצפנה א-סימטרית
- Sha-2 עבור hash חד כיווני

- עבור יצירת מספרים רנדומאליים יש להשתמש במנגנון מבוסס crypto random generator

הגנה על מפתח\ות הצפנה:

- יש לאבטח את מפתח \ מפתחות ההצפנה הנמצאים בשימוש המערכת מפני גישה \ שימוש זדוני ללא הרשאה בהתאם לסוג המפתח – ציבורי \ פרטי.
- יש להגן על המפתח מפני הרס או שינוי בצורה לא מורשת.
- יש לנהל בקרה ודיווח לגבי ביצוע גישות ושימוש במפתחות הצפנה.
- יש להבטיח יכולות שיחזור וגיבוי בשימוש במפתחות הצפנה (כדי להבטיח שיהיה ניתן לשחזר מידע רגיש שהוצפן עם מפתח שאבד).
- על המערכת להימנע משמירת מידע רגיש בקובצי הגדרות, קבצים זמניים, cookies, זיכרון מטמון וכו'. במידה ומידע נשמר במקומות אלו, נדרש לוודא כי לאחר סיום עבודה במערכת מידע שיעורי זה ימחק.

3.10 ניהול שגיאות

- הודעות שגיאה שיוצגו למשתמש כתוצאה משגיאות המתרחשות באפליקציה יהיו הודעות *שאין בהן כדי לחשוף את אמצעי האבטחה במערכת*.

- הודעות שגיאה שיוצגו למשתמש יהיו הודעות ***<u>שאין בהן כדי לחשוף את התשתית האפליקטיבית לגרסאותיה השונות</u>*** כגון: מערכות הפעלה, שרתי web, שרתי אפליקאציה, בסיסי נתונים, פרוטוקולים בשימוש, Web Services בשכבות נמוכות וכדומה.
- אין להציג כל מידע פנימי (כולל: כתובות IP, שמות שרתים, נתיבים של קבצים במעה"פ, שמות משתמשים, מספרי אשראי, סיסמאות, מפתחות הצפנה וכו') בהודעות שגיאה המוצגות למשתמש.
- ***<u>יש לוודא כי הודעות שגיאה אינן חושפות שום מידע רגיש בנוגע למבנה המערכת ומשאבי המערכת. הודעות השגיאה שיוצגו יהיו ערוכות ויציגו מידע כללי.</u>***
- כאשר קלט המשתמש אינו מתאים לתבנית הנדרשת בשדה קלט, יש להציג למשתמש הודעת שגיאה המפרט מהי התבנית בה נדרש להשתמש.
- על המערכת לנהל מערך ללכידת שגיאות בזמן ריצה :
  - יש לצפות שגיאות מראש וללכוד אותן בקוד המערכת.
  - בשגיאות שהוגדרו כשגיאות כתוצאה מפעילות הקשורה באבטחת מידע יש לנהוג לפי מה שהוגדר במידול הסיכונים של המערכת, כולל דיווח למנהל המערכת, חסימת משתמש וכו'.
- יש לדאוג לכך שמידע משגיאות יהיה מתועד ע"י המערכת בדפי ה log שלה.
- על המערכת להתמודד עם שגיאות בהיבט של זמינות כך שאם למשתמש מסוים מתרחשת שגיאה הוא אינו חוסם גישה למשתמשים אחרים שמריצים את המערכת (קריסה כללית).
- במערכות רגישות ובסיכון בינוני ומעלה, המערכת תכלול יכולת לאחזור הודעות שגיאה (אחזור מלא, אחזור חלקי לפני פרמטרים שונים).
- פרמט הדיווח של הלוגים צריך להתאים לפורמט מערכת SIM \ SOC כך שיהיה ניתן לאסוף את הודעות השגיאה.

3.11 חיווי ובקרה

- האפליקציה תתעד את הנתונים הבאים, במידה והם מוגדרים, עבור כל פעולה במערכת:
  - Timestamp.
  - זיהוי המשתמש.
  - מיקום המשתמש (מחשב/IP).
  - מיקום המשתמש במערכת (מסך, טופס, טבלה וכדומה).
  - פרטים מלאים של הפעולה המבוקשת.
- בנוסף יתועדו הפעולות הבאות:
  - צפייה במידע במערכת.
  - עדכון מידע במערכת.
  - כתיבה ומחיקה של מידע במערכת.
  - כל פעולות הניהול במערכת.
  - כל פעולות הזיהוי במערכת, כולל כישלונות של פעולות אלו והסיבה לכך.
  - כל פעולות ההרשאות במערכת, כולל כישלונות של פעולות אלו.
  - שגיאות מערכת.
  - ועוד, בהתאם לצורך.
- התיעוד יתבצע בשתי שכבות: תיעוד פעולות משתמשי מערכת באפליקציה, תיעוד גישה לנתוני המערכת בבסיס הנתונים.
- חשוב להדגיש כי התיעוד לא יכיל את נתוני הזיהוי של משתמשים או נתונים רגישים אשר שמורים בבסיס הנתונים של המערכת.
- כל פעולת תיעוד, בכל הרמות של המערכת, חייבת להכיל את המשתמש המבצע את הפעולה בפועל על מנת למנוע התכחשות משתמשים לפעולותיהם.

מעקב:

- יש לוודא כי נתוני התיעוד והמעקב נשמרים באופן מאובטח במערכת.
- יש לוודא כי רישומי התיעוד אינם נגישים למשתמשים ללא הרשאות מנהל מערכת.
- יש לוודא כי נתוני התיעוד מגובים יחד עם שאר נתוני המערכת.
- יש לקיים מנגנון ארכיב לנתוני תיעוד ישנים אשר אינם נחוצים לשם פעולתה התקינה של המערכת.
- יש להגדיר בתיאום עם מזמין המערכת את תקופת שמירת נתוני התיעוד של המערכת.

3.12 חתימת קבצים

- יש לבצע שימוש ב strong name ולחתום את קוד הפרויקט לאחר יצירת גרסת ייצור יציבה.
- מומלץ להחליף חתימה זאת בכל שחרור של גרסה חדשה.

CAS 3.13

- מומלץ לממש מנגנון CAS במערכת על מנת להגביל את גישת האפליקציה למקורות מידע שלא נדרש לבצע אליהם גישה כגון FTP, Unmanaged code וכו'.

3.14 ניהול הגדרות

- על המערכת לפרט באפיון את כל אמצעי גישות ניהול ההגדרות שבמערכת:
- יש להגדיר תחת איזה חשבון רצה המערכת (משתמש, מנהל, חשבון מערכת ..), הדרישה היא כי המערכת תרוץ תחת חשבון עם רמת הרשאות הנמוכה ביותר הניתנת כך שלא תאפשר ביצוע פעולות לא רצויות ע"י משתמשים רגילים. לא יופעל שום רכיב עם זיהוי SYSTEM ו/או הרשאות ADMIN או מקבילות להם.
- יש להגדיר דרכי גישה מאובטחות למשאבים חיצוניים כגון בסיס מידע, מערכת קבצים וכו' (למשל ע"י הצפנת מחרוזת קישור לבסיס מידע).
- יש להגדיר גישה מאובטחת לאדמיניסטרציה במערכת כולל זיהוי חזק והגבלת הגישה למורשים בלבד. יש לשקול הגדרת כתובות IP מסוימות שרק מהן ניתן לגשת לממשק הניהול.
- יש לדאוג לכך שלא יהיה ניתן לחשוף\לגשת למידע רגיש הקיים בקובצי הגדרות למשתמשים ללא הרשאות מתאימות.
- אין לשמור מידע רגיש בקובצי הגדרות. במידה ונדרש יש להצפין אותו ע"י שימוש ב dpapi.
- יש לתת דגש לתהליך זרימת המידע, ולהתאים מצבים אפשריים באתר לדפים. כך שאם גישה לדף מסוים מותנית בביצוע סדרת פעולות עוקבות לא יתאפשר דילוג על אחת הפעולות. כל דף משויך למצב. למשל הגשת בקשה לפתיחת חשבון תתאפשר רק אם המשתמש מילא סדרה של טפסים, ללא יכולת לדלג על אחד השלבים בדרך.

3.15 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 (עפ"י האפיון שימוש ברשימות white list וב regular expression לפני הכנסת קלט למערכת).
- יש לבדוק שטיפוס הקלט המתקבל הוא מהסוג המצופה.
- יש לבדוק כי טווח הערכים שמתקבל מתאים להגבלות שנקבעו.
- יש לבדוק את הרכב התווים בקלט, ולוודא שהוא אינו מכיל תווים אסורים. ככלל יש להימנע ככל האפשר מקבלת קלט שאינו מכיל ערכים אלפא נומריים, למעט רווחים.
- יש לוודא כי ערכו של הקלט תואם ללוגיקה העסקית של רכיב היעד.
- יש לוודא כי הקלט ב encoding המתאים 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 בפרויקט נפרד \ תשתית .
- בכל מקרה יש לבצע סינון מסודר של תווים למניעת הזרקת שאילתות SQL.
- כל תעבורת השאילתות תבוצע ע"י שימוש ב- stored procedures באופן הנכון וללא שימוש בהעברת פרמטרים בקריאה ל- stored procedure.

מניעת מתקפות חסימת שירות

- יש לקחת בחשבון את כלל האיומים העלולים לגרור מתקפות Denial of service (מניעת שירות) ולגבש בקרות כנגדם.
- במנגנון נעילת משתמשים יש לקחת בחשבון את אלמנט הזמינות כך שיהיה ניתן לשחרר משתמש שננעל בצורה מהירה יחסית.

הגנה מפני buffer overflow

- יש לאמת פרמטרי מחרוזות כקלט ופלט – יש לוודא את אורך המחרוזת שלא תחרוג מהמקסימום.
- יש לאמת גבולות של מערכים.
- יש לאמת אורך נתיב לקבצים.

הגנה מפני מתקפות Network eavesdropping

- הצפנת תווך תקשורת\ הודעה בזמן ביצוע הזדהות.
- הצפנת תווך תקשורת \ הודעה בזמן העברת מידע הקשור בפרטי זיהוי משתמש כגון החלפת סיסמא וכו'.

הגנה מפני מתקפות Brute force & Dictionary attacks

- מימוש מדיניות סיסמאות חזקה.
- מימוש מנגנון נעילת משתמשים.
- שמירת סיסמאות ע"י שימוש ב hash בתוספת מספר רנדומאלי.

הגנה מפני מתקפות session hijacking ו- session replay

- אין לאפשר פתיחה של יותר מ session אחד עבור משתמש (במערכות רגישות בלבד)
- יש לבצע בדיקות אימות ל session לפני מתן גישה כלשהי.
- שימוש לבצע שימוש בתווך מוצפן כדי שלא יהיה ניתן לגנוב cookie המועבר לאפליקציה.
- למניעת מתקפות replay יש ליצור ערך חד ערכי עבור כול הודעה הנשלחת, כמו כן מומלץ לשלב חתימה בגוף ההודעה – timestamp.

מניעת שמירת נתונים במטמון הדפדפן

- אופציית אחסון הדפים (Caching) תהיה מבוטלת עבור כל הדפים באפליקציה ולכל סוגי הדפדפנים.

מניעת חשיפת תוכן תיקיות השרת

- יש לבטל את מאפיין ה-Directory Listing בכל אחת מהתיקיות הוירטואליות על שרת האפליקציה.

מניעת אפשרות אחסון פרטי הזדהות בדפדפן

- יש לבטל אפשרות ה- Password Auto complete ע"י שליחת מאפיין מתאים בתגי ה-Password וה-Form בדף ה-HTML. דוגמא:

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

מניעת גישה לדפי בדיקות ודפים שאין אליהם קישורים נדרשים באפליקציה

- יש למנוע גישה לדפי סביבת הבדיקות ולהסיר אותם מסביבת הייצור של המערכת.
- יש לפעול לפי נוהל העברה ליצור מסודר.