נספח 2.7 – דרישות אבטחת מידע

1. מבוא 3

1.1. מטרה 3

1.2. אחריות 3

2. סיכונים עיקריים 4

3. דרישות הזדהות והרשאות 5

3.1. הרשאות 5

3.2. הפרדת מידע – טוקניזציה 7

3.3. בקרת גישה 7

3.4. הגנה על מידע 7

4. ארכיטקטורה 9

4.1. ניהול Session 9

5. פיתוח ותחזוקה 10

5.1. תהליך הפיתוח 10

5.2. קונפיגורציות וקוד פתוח (Open Source) 11

5.3. הפרדת סביבות 11

5.4. בדיקות 11

5.5. פיתוח מאובטח 12

5.5.1. בדיקת קלט 12

5.5.2. הגנה על הדפדפן 12

5.5.3. קריפטוגרפיה 13

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

5.6. טיפול בשגיאות ותיעוד 14

6. תפעול ותחזוקה 15

6.1. הגנה פיזית 15

6.2. הגנה לוגית 15

6.3. קבצים ומשאבים 15

6.4. שמירת עדכניות 16

7. היפרדות 17

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

1. מבוא

1.1. מטרה

מטרת המסמך היא להגדיר למציע המתמודד במכרז לפיתוח מערכת הגשת בקשות וניהול מלגות, את דרישות אבטחת המידע שבהן על המערכת לעמוד. המסמך יסייע בקביעת אותם כללים מנחים שיהיה על הספק לפעול לפיהם בעת פיתוח המערכת, ובכך לעמוד בדרישות אבטחת המידע של הארגון.

1.2. אחריות

דרישות אבטחת המידע יהיו חלק בלתי נפרד מתהליכי התכנון, האפיון והפיתוח, הבדיקות, הטמעת המערכת ותחזוקתה, ועל כן האחריות לעמוד באותן דרישות מוטלת על הספק.

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

2. סיכונים עיקריים

בטבלה למטה מרוכזים עיקרי סיכוני אבטחת המידע. המענה לסיכונים אלה מפורט בהמשך. המציע מוזמן להציע הצעות נוספות/אחרות להפחתת הסיכונים.

האיום	השפעה	סבירות	פירוט האיום והשלכותיו
השחתת מידע	בינונית	נמוכה	גורם עוין, או לחלופין גורם בעל עניין בתהליך המלגות, עלול לשנות את המידע שהגיע או נצבר כדי להקל/להקשות את אישור המלגה.
גנבת מידע	גבוהה	בינונית	גורם עוין עלול לגשת למידע ולפרסם אותו לצרכים שונים.
הפרעה לשימוש שוטף (DoS), כולל כופרה	בינונית	גבוהה	גורם עלול לנסות להשבית את המערכת באמצעים שונים כדי לגרום נזק תדמיתי או כדי להרוויח כסף (כופרה).

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

3. דרישות הזדהות והרשאות

3.1 הרשאות

3.1.1.1. מימוש הזדהות לכל הדפים/מסכים/API – כניסה למערכת דורשת אימות באמצעות סיסמה.

3.1.1.2. רמות הרשאות – יתאפשר להגדיר כמה רמות של הרשאות, על פי אפיון האקדמיה.

3.1.1.3. עבור סוגי המשתמשים האלה יש ליישם פתרון הזדהות המממש 2FA באמצעות קוד חד-פעמי המתקבל למכשיר הנייד של המשתמש:

- רקטור אוניברסיטה או נציגו, כפי שיוגדר באקדמיה
- שופט
- יו"ר ועדת שיפוט
- מועמד
- מלגאי
- נדחה

3.1.1.4. שדות סיסמה – על שדות להזנת סיסמה להיות ממוסכים ולעולם לא לחשוף את תווי הסיסמה של המשתמש.

3.1.1.5. יישום מדיניות סיסמאות חזקה – יש לאכוף דרישה לאורך מינימלי של שמונה תווים לכל הסיסמאות של משתמשים רגילים, ועשרה תווים עבור משתמש בעל הרשאות ברמה גבוהה יותר. כמו כן חובה שהסיסמאות יכילו לפחות שלוש מקבוצות התווים האלה:

- אותיות גדולות (A, B, C, D וכו').
- אותיות קטנות (a, b, c וכו').
- ספרות (0, 1, 2, 3, 4 וכו').
- סימנים מיוחדים ($, #, @, ! וכו').

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

3.1.1.6. מדיניות תוקף לסיסמאות – המערכת תחייב את המשתמשים להחליף סיסמאות:

- תוקף סיסמה יפוג לאחר 90 יום לכל היותר.
- סיסמה לא תחזור על עצמה במהלך ארבעה דורות.

3.1.1.7. מנגנון לשינוי פרטי הזדהות – יש לוודא כי תהליך שינוי הסיסמה כולל את הסיסמה הישנה, את הסיסמה החדשה ואישור הסיסמה. נוסף על זה, על תהליך שחזור הסיסמה שנשכחה ועל נתיבי שחזור אחרים להימנע מחשיפת הסיסמה הקיימת או החדשה.

3.1.1.8. תעבורה מוצפנת – על פרטי ההזדהות המועברים לשרת לעבור בתווך מוצפן.

3.1.1.9. מניעת חשיפה של שם משתמש – על הודעות השגיאה להיות כלליות ולא לחשוף מידע, למשל בתהליכי איפוס סיסמה, שחזור סיסמה או במנגנון ההזדהות עצמו. כלומר, המערכת לא תציין איזה שדה שגוי (שם משתמש, סיסמה וכו').

3.1.1.10. סיסמאות ברירת מחדל – אין לאפשר סיסמאות ברירת מחדל.

3.1.1.11. הגנה מפני מתקפות Brute force – לצורך חסימה ומניעת מתקפות אוטומטיות או מתקפות נפוצות נגד תהליך האימות, ובהן מתקפות Brute-Force או מתקפות מניעת שירות (DoS – Denial of Service), יש להשתמש במנגנוני anti-scripting, כגון CAPTCHA.

3.1.1.12. הגנה והצפנה של פרטי הזדהות – נדרש כי כל פרטי האימות המשמשים לכניסה למערכת יהיו מוצפנים ויישמרו במקום מוגן.

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

3.2. הפרדת מידע – טוקניזציה

3.2.1.1. טוקניזציה – יש להפריד את המידע למפתח ולמידע. יש לשמור את המפתח (למשל מת"ז) עם טבלת המרה למפתח פנימי שאינו קשור למפתח המקורי. כלומר, אין להשתמש בפונקצייה מתמטית על המפתח המקורי, אלא לשייך לו מספר אקראי וייחודי (חד-חד-ערכי – **unique**). טבלה זו תישמר מוצפנת, ורק התוכנה תוכל לגשת אליה. כלומר, אין לאפשר גישה ידנית.

3.2.1.2. הפרדה – כל שאר המידע יקושר אך ורק למשתנה הפנימי.

3.2.1.3. תצוגה – המפתח החיצוני יוצג גבי מסך ובדוחות רק חלקית, שתי ספרות ראשונות ושתי ספרות אחרונות. רק בדוחות ספציפיים עבור משתמשים ספציפיים יוצג המפתח המלא. משתמשים ודוחות אלה יוגדרו באפיון.

3.3. בקרת גישה

3.3.1.1. תהליך אישור של פונקציות ושירותים שונים – יש לוודא כי עקרון ההרשאות המינימליות (Least Privilege) מתקיים עבור כל המשתמשים. המשתמשים צריכים להיות בעלי הרשאות גישה רק לפונקציות, לקובצי נתונים, לכתובות URL, לבקרים (Controllers), לשירותים (Services) ולמשאבים אחרים אשר יש להם הרשאה ייעודית אליהם. העיקרון מיישם הגנה מפני זיוף או העלאת הרשאות לא מורשות.

3.4. הגנה על מידע

3.4.1.1. מסמכים רשמיים סרוקים הם בבחינת סיכון גבוה. לכן אם מגיש הבקשה מצרף מסמכים, תשאל המערכת את הגורם המטפל אם המסמך הכרחי להמשך הטיפול. אם לא, המסמך יימחק.

3.4.1.2. אם המסמך הכרחי הוא יישמר מוצפן. המערכת תזכיר למשתמש שאישר לכלול מסמך סרוק במערכת, לאחר זמן שיוגדר באפיון, או לאחר סיום פעילות שתוגדר באפיון, שיש להסיר את המסמך מהמערכת.

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

.3.4.1.3 מידע רגיש אינו נשמר ב-Cache – יש לוודא כי כל הדפים והמסכים המכילים מידע רגיש אינם נשמרים ב-Cache של צד הלקוח, כולל מנגנונים להשלמה אוטומטית.

.3.4.1.4 מידע רגיש אינו נשלח ב-URL – את כל המידע הרגיש יש לשלוח לשרת בתוך גוף או בכותרת הודעת HTTP, כלומר פרמטרי URL לעולם אינם משמשים לשליחת נתונים רגישים.

.3.4.1.5 אין לשמור מידע רגיש בצד לקוח – יש לוודא כי המידע השמור בצד לקוח, כגון Cookies או מידע הנשמר ב-Session/Local Storage, אינו מכיל מידע רגיש.

.3.4.1.6 מידע רגיש "מנוקה" במהירות מהזיכרון – יש להקפיד שהמידע הרגיש "ינוקה" מהזיכרון ברגע שלא יהיה נחוץ עוד, ויטופל לפי פונקציות וטכניקות הנתמכות ב-Framework / מערכת הפעלה.

.3.4.1.7 יש להגן על תווך התקשורת באמצעות תקן ההצפנה TLS 1.2 בכל החיבורים הרלוונטיים. יש להגן על תווך התקשורת באמצעות TLS בכל תווכי התקשורת שעובר בהם מידע רגיש כגון סיסמאות, מידע עסקי וכד'.

.3.4.1.8 מחיקת מידע: לכל פריט מידע יוגדר משך החיים שלו – במונחים של זמן מאז הוקם או במונחי פעילות על המידע. בתהליך האפיון יוגדר לכל פריט מידע אם יש למוחקו ומתי או בעקבות איזה אירוע. בתהליך המחיקה תציג המערכת את המידע העומד להימחק, תאפשר למשתמש מורשה לבטל מחיקה של פריטים מסוימים, ואז תבצע מהלך מחיקה לשאר הפריטים. זהו תהליך עיתי, ובמהלך האפיון יוגדר כל כמה זמן יבוצע.

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

4. ארכיטקטורה

4.1. ניהול Session

4.1.1.1. תוקפו של ה-Session פג בעת יציאת המשתמש – יש לוודא שה-Session מבוטל כאשר המשתמש יוצא או מתנתק מהמערכת.

4.1.1.2. ה-Session מסתיים לאחר חוסר פעילות – יש להטמיע התנתקות קבועה מראש לאחר 15 דקות של חוסר פעילות, שלאחריו יופעל מנגנון ניתוק תקשורת שיחייב זיהוי מחדש של המשתמש.

4.1.1.3. אם מנגנון הניתוק מטיל מגבלה על פעילות בעלת אופי רציף, יש להתריע למשתמש לפני ניתוק התקשורת.

4.1.1.4. Session/Token הם ארוכים ואקראיים במידה מספקת – על המזהה (ID) של ה-Session או טוקנים להיות ארוכים, אקראיים וייחודיים.

4.1.1.5. Session Cookies הם בעלי הגדרות מאובטחות – יש לוודא כי Session Tokens המוגדרים ב-Cookies לצורך אימות מוגדרים בתכונות: "HttpOnly" ו-"Secure".

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

5. פיתוח ותחזוקה

5.1 תהליך הפיתוח

הספק יפתח את המערכת עבור האקדמיה הלאומית הישראלית למדעים. תהליך הפיתוח יהיה מותאם לתהליך SSDLC (Secured Software Development Life Cycle) מקובל, אשר יכלול ביצוע הבדיקות האלה:

5.1.1.1. Design Review – אפיון המערכת ותכנונה יסוכמו בפגישה או בסדרת פגישות, לפי הצורך, שבהן יציג הספק לאקדמיה את האפיון, והאקדמיה תאשר את התאמתו לצרכיה. במסגרת אפיון זה יוצגו ויאושרו גם היבטי אבטחת המידע של המערכת.

5.1.1.2. Code Review – בדיקת קוד (Security Code Review) לקוד המקור של המערכת לקראת סיום פיתוח המערכת. <מה יבוצע? הבדיקה? אם כן: הבדיקה תבוצע באמצעות... אם לא הבדיקה לכתוב מה יבוצע> יבוצע באמצעות כלים אוטומטיים, בידי הספק ועל חשבונו. בדיקה זו תתבצע לכל גרסה בעלת חשיבות.

5.1.1.3. Security Assessment – הבדיקה תתבצע בידי הספק ועל חשבונו. היא תכלול גם מבדק חדירה אחרי התקנת המערכת בסביבת הבדיקות ולפני העברתה לסביבת הייצור, לכל גרסה בעלת חשיבות.

5.1.1.4. הספק מתחייב לשתף פעולה עם האקדמיה ועם גורמים מטעמה כאשר זו תבצע מבדקי חדירה בסביבת הייצור, באופן שוטף. אם תימצאנה תקלות בבדיקות החדירות, על הספק לפעול לתקנן במהירות האפשרית וכחלק מתנאי התחזוקה (ללא תשלום נוסף).

האקדמיה שומרת לעצמה את הזכות לצרף אנשי מקצוע מטעמה לבדיקות אלה, או לבצע מטעמה בדיקות אלה ונוספות. אם תתגלינה בעיות בבדיקות אלה, הן תתוקנה בידי הספק ועל חשבונו.

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

5.2. קונפיגורציות וקוד פתוח (Open Source)

5.2.1.1. גרסאות וקונפיגורציות – יש לוודא כי כל הרכיבים, קובצי ההגדרות וה-Framework שעליהם מבוססת המערכת מוגדרים על פי המלצות היצרן ומעודכנים לגרסה האחרונה שעליה ממליץ היצרן. חובה לשנות את סיסמאות ברירת המחדל (אם יש כאלה),

5.2.1.2. רכיבי צד שלישי – יש לנהל רשימה של רכיבי צד שלישי, כולל קוד פתוח (open source), אשר כלולים בקוד. יש לבדוק אחת לחודש פגיעויות מוכרות ולעדכן לגרסאות החדשות ביותר. העדכון בסביבת הייצור יבוצע רק לאחר בדיקה מקדימה ויסודית של העדכון והשפעתו על המערכת, ולאחר מכן יתואם עם האקדמיה המועד המדויק לביצוע העדכון.

5.3. הפרדת סביבות

הספק יספק לפחות ארבע סביבות עבודה:

5.3.1.1. סביבת פיתוח, אשר תשמש לפיתוח ולבדיקות ראשוניות של גרסאות חדשות.

5.3.1.2. סביבת בדיקות, אשר תשמש לבדיקות מקיפות של גרסאות חדשות או שינויים לגרסה האחרונה.

5.3.1.3. סביבת בדיקות זהה לייצור (staging), אשר תשמש לשחזור בעיות בסביבת הייצור ולבדיקות תאימות (regression testing) של שינויי תשתית, כגון מערכת ההפעלה, open source וכו'.

5.3.1.4. סביבת ייצור.

5.4. בדיקות

5.4.1.1. מידע לבדיקות – בסביבת הבדיקות ייעשה שימוש במידע אמיתי משובש שיופק בידי הספק באופן שלא תיתכן אפשרות לשחזור המידע האמיתי.

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

5.4.1.2. בסביבת ה-staging ייעשה שימוש במידע שיועתק מגיבוי המידע העדכני ביותר.

5.4.1.3. סביבת הבדיקות – סביבות הפיתוח והבדיקות תהיינה מופרדות מסביבת הייצור. לא תתאפשר גישה ישירה מסביבות הפיתוח והבדיקות לסביבת הייצור ולהפך. העברת גרסה מסביבת הבדיקות לייצור תיעשה רק באמצעות תוכנה.

5.5. פיתוח מאובטח

5.5.1. בדיקת קלט

5.5.1.1. בדיקה של כל נתוני הקלט – יש לאמת את כל הפרמטרים בצד השרת, בין שמקורם בקלט שהמשתמש הזין ובין שאלה נתונים שנשלחו לצד הלקוח באמצעות האפליקציה וחזרו לצד השרת. יש לבדוק אם פרמטרי חובה לא נשלחו. בדיקות התקינות יכללו בדיקות ביטויים רגולריים (Regular expression) ובדיקות נוספות על פי אפיון האקדמיה.

5.5.1.2. בדיקות תקינות נוספות – במקומות שבהם הדבר רלוונטי, יש לבצע גם בדיקות תקינות נוספות, למשל ספרת ביקורת כחלק ממת"ז, מס' חשבון בנק, כרטיס אשראי וכדומה, ובכל מקום שבו יש כזו, על הספק למצוא את האלגוריתם המתאים לכל שדה רלוונטי, כולל שדות שמקורם בחו"ל.

5.5.1.3. קידוד או Escaping של פלט – במערכות Web יש לוודא שכל הנתונים המועברים לצד הלקוח מקודדים באמצעות HTML Encode כדי להבטיח שהאפליקציה אינה חשופה להתקפות XSS (Cross Site Scripting).

5.5.2. הגנה על הדפדפן

5.5.2.1. הגנות מפני CSRF (Cross Site Request Forgery) – על האפליקציה ליישם מנגנונים למניעת CSRF, כגון Anti-CSRF Tokens, באמצעות טוקנים חזקים ואקראיים, או להשתמש לצורך כך בכללי הפיתוח המאובטח המונעים CSRF.

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

5.5.3. קריפטוגרפיה

5.5.3.1. שימוש באלגוריתם הצפנה חזק דו-כיווני – יש להשתמש באלגוריתמי הצפנה RSA/AES להצפנה ולפיענוח של מידע, כגון connection string, סיסמאות למשאבים פנימיים כמו בסיס נתונים, וטבלת ההמרה בין זיהוי המלגאי לבין המפתח הפנימי שלו. יש להשתמש באלגוריתם ה-RSA הן להצפנת המפתח הציבורי וטבלת ההמרה והן לחתימות דיגיטליות – אם יוחלט בעצה אחת עם האקדמיה שיש צורך בכאלה.

5.5.3.2. הגנה על רשומות סיסמת המשתמש בבסיס הנתונים – על המערכת לשמור סיסמאות משתמשים בצורת Hash (גיבוב) באמצעות שימוש באלגוריתם חד-כיווני, ייעודי למטרה זו, למשל: bcrypt, scrypt או PBKDF#2, ולפי הרגולציה "ההוראה לניהולי סיכוני אבטחת מידע של הגופים המוסדיים", משרד האוצר, סעיף 3.5 ה': סיסמאות לא יישמרו באופן גלוי (Clear Text) או באופן הניתן לשחזור ברשומות, בזיכרון או במאגרי מידע.

5.5.3.3. שימוש ב-Salt – יצירת Hash של סיסמה צריך לכלול ערך Salt כדי למנוע תקיפות מסוג Rainbow Table. יש לשלב את Salt עם ערך הסיסמה לפי אלגוריתם הגיבוב (hashing) המתאים, לפני ביצוע גיבוב (hashing) על ערך ה-Salt. יש להשתמש ב-Salt ייחודי ואקראי לכל חשבון משתמש, ובאורך 64 סיביות (8 תווים) לפחות. ערך ה-Salt אמור להיות ארוך יותר עבור נתונים קצרים כמו קודי PIN או נתונים רגישים במיוחד. הערך עשוי להיות מאוחסן בטקסט (Clear-text) לצד ערך הסיסמה המגובב (hashed), באותה רשומת חשבון משתמש. לחלופין ניתן להשתמש בשם המשתמש כערך ה-Salt לכל משתמש.

5.5.3.4. שימוש באלגוריתמי Hash חזקים – נכון להיום אלגוריתם ה-Hash החזק ביותר הוא SHA3-512. אין מניעה להשתמש גם ב-SHA2-256.

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

5.6. טיפול בשגיאות ותיעוד

5.6.1.1. הודעות שגיאה – יש לוודא כי האפליקציה אינה מציגה הודעות שגיאה פרטניות או Stack Trace המכילות מידע רגיש העלול לסייע לתוקף, כולל SessionID, גרסאות תוכנה או Framework ומידע אישי.

5.6.1.2. מנגנון תיעוד ובקרה – על כל ניסיונות הכניסה הכושלים וכל אירועי האבטחה להירשם בלוג אפליקטיבי, לצורך תיעוד במערכת. יש לדווח על ניסיונות כניסה רצופים לאנשי האבטחה המתאימים ולמערכת ה-SIEM (Security Information and Event Management) הארגונית של הספק. יש לתעד גם ניסיונות כניסה מוצלחים לצורך סיוע לתהליך פורנזיקס (Forensics) – זיהוי מקורות הבעיה – לצורך טיפול אבטחתי ומשטרתי.

5.6.1.3. בכל רישום ביומן אירועי אבטחה יש לציין את המידע הנדרש לניתוח האירוע, כולל (אך לא מוגבל לאלה):

- מקור האירוע (כתובת ה-IP).
- המשתמש הגורם לאירוע.
- סוג האירוע.
- השעה והתאריך שבהם אירע האירוע.
- המידע או הרשומה שהושפעה, או הפניה למידע זה.

5.6.1.4. לוגים של האפליקציה אינם כוללים מידע רגיש – אסור שמידע רגיש (כמו סיסמאות, מזהי Session וכדומה) יישמר בתוך הלוג. יש לכתוב את שם המשתמש ואת כתובת ה-IP שיזמו את האירוע.

5.6.1.5. כל גישה למידע רגיש תתועד בלוג – יש לוודא כי הגישה למידע רגיש מתועדת בלוג, או כאשר נדרש רישום גישה. מידע רגיש יוגדר עם האקדמיה יחד בשלב האפיון.

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

6. תפעול ותחזוקה

6.1. הגנה פיזית

6.1.1.1. מבנה – על השרתים (אם בחצרי הספק ואם אלה המספקים שירותי ענן) להיות מוגנים פיזית, באמצעות שמירה 7*24.

6.1.1.2. גישה – גישה לשרתים תתאפשר רק למורשים ובאמצעות שתי מערכות הגנה בלתי תלויות (למשל כרטיס זיהוי וקוד אישי של בעל כרטיס הזיהוי).

6.1.1.3. צילום – סביבת השרתים תנוטר 7*24 באמצעות צילום.

6.2. הגנה לוגית

6.2.1.1. FW – השרתים יוגנו באמצעות FW למניעת חדירת וירוסים דרך הרשת.

6.2.1.2. IPS (Intrusion Prevention System) – סביבת הייצור תוגן באמצעות מערכת למניעת פריצות.

6.2.1.3. בקרה וניתוח – בסביבת הייצור תהיינה מערכות SIEM לאיסוף ההתרעות וכדי לאפשר ניטור, מניעת הפרעות בזמן אמת ותחקור אירועים באמצעות ה-SOC (Security Operation System).

6.3. קבצים ומשאבים

6.3.1.1. גיבויים – נוסף על גיבוי רגיל ברשת, יש לשמור גם גיבוי פיזי מנותק מהרשת, בתדירות המאפשרת שחזור המידע במקרה של התקפת כופרה. תדירות הגיבויים תיקבע באקדמיה. יישמרו לפחות שלוש גרסאות אחרונות בכל רגע נתון.

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

6.3.1.2. הגנה מפני הפניות לא מאובטחות – יש לוודא שהפניות ה-URL והעברות שונות מאפשרות רק יעדים מוגדרים ב-whitelist, ושלא ניתן לגשת ישירות לקבצים רגישים בפנייה ישירה.

6.3.1.3. סריקת אנטי-וירוס והלבנה לקבצים – על קבצים המתקבלים ממקורות חיצוניים לעבור אימות של סוג הקובץ האמור להתקבל, ולהיסרק באנטי-וירוס ובמוצרי הלבנה, כדי למנוע העלאת תוכן זדוני.

6.4. שמירת עדכניות

6.4.1.1. סביבת בדיקות זהה לייצור – הספק יתחזק סביבת בדיקות המכילה תמידית את גרסת הייצור (staging) כדי לאפשר בדיקת בעיות וכדי לאפשר בדיקות רגרסיה כאשר יש צורך לעדכן גרסאות מ"ה, בסיס נתונים או קוד פתוח (תוכנות תשתית), אם וכאשר מתגלות בהן תקלות אבטחה.

6.4.1.2. עדכון תשתית הלוקה בבעיית אבטחה – הספק מתחייב לעדכן תשתיות תוכנה לא יאוחר משלושה חודשים לאחר גילוי תקלת אבטחה חמורה לפי הגדרותיו של ספק תשתית התוכנה כאמור.

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)

7. היפרדות

עם קבלת הודעה מהאקדמיה על סיום התקשרות על הספק לבצע את האמור בפסקה זו, נוסף על ביצוע האמור בפסקת ההיפרדות במסמכי המכרז:

7.1.1.1. להעביר את כל התיעוד ומידע אבטחת המידע לספק החדש באופן מסודר ולהקצות לכך לפחות שבוע עבודה רצופה על חשבונו.

7.1.1.2. למחוק את כל המידע השמור בחצרו, כאשר האקדמיה תבקש זאת.

7.1.1.3. לספק הצהרה חתומה בידי עורך דין שאכן כך נעשה.

קראנו, הבנו, מקובל עלינו _____

(חתימות בראשי תיבות בצירוף חותמת תאגיד)