הצפנת נתונים בתעבורה ובמנוחה - מסמך מדיניות

גרסה 1.1

מעקב גרסאות

מס"ד	תאריך	עודכן על ידי	תיאור השינויים
1.0	13.12.2021	אופיר יהב	גרסא ראשונה
1.1	13.12.2022	אופיר יהב	תיקוף ועדכון המסמך. עדכון הלוגו לאחר השינוי הארגוני.

נתוני גרסת המסמך

גורם	תפקיד	שם מלא	תאריך	חתימה
נערכה ע"י	ראש תחום מתודולוגיות הגנה בסייבר	אופיר יהב	13.12.2022	(חתימה)
נבדקה ע"י	ראש תחום אבטחת אפליקציות	אברהם ברדה	13.12.2022	(חתימה)
נבדקה ע"י	ארכיטקט מערך ההגנה בסייבר	שרון משולם	13.12.2022	(חתימה)
אושרה ע"י	מנהל מערך ההגנה בסייבר	חגי פרלמוטר	13.12.2022	(חתימה)

תוכן עניינים

1. כללי

1.1 יחידת מערך הדיגיטל הלאומי השייכת לרשות התקשוב הממשלתי עוסקת בייזום, פיתוח ותפעול תשתיות ופתרונות טכנולוגים מאובטחים עבור גופי הממשלה, בין היתר במטרה להנגיש את שירותי הממשלה לאזרחים ועסקים ולשם חיבור גופי הממשל השונים לשירותים מקוונים ולשירותי מובייל מותאמים.

1.2 במטרה להגן על נכסי המידע המאוחסנים בתשתיות מערך הדיגיטל הלאומי, מקפיד הארגון על כל מרכיבי אבטחת המידע – חסיון המידע, זמינות המידע ושלמות המידע.

1.3 אחת הדרכים לשמירה על מרכיבי אבטחת המידע הינה הצפנת המידע – הן במנוחה (At-Rest - בשטחי האחסון של המידע) והן בתעבורה (In-Transit - בתנועה בין רכיבי התשתית ומערכותיה).

1.4 מסמך זה מפרט סט הנחיות כלליות להצפנת המידע בתעבורה ובמנוחה והוא מהווה נספח למסמך מדיניות אבטחת המידע בכלל ומדיניות פיתוח מערכות מאובטחות בפרט.

1.5 הנהלת הארגון תתווה את קווי המדיניות בנושא הצפנת נתונים בתנועה ובמנוחה ותעביר אותה אל דרגי הביצוע בנהלים פנימיים של הארגון – נהלים אשר יופצו בארגון ויפורסמו בפורטל הארגוני.

1.6 נהלי הארגון יכללו הוראות וכללים אשר מטרתם להגן על מידע הנמצא בתנועה ובמנוחה – כגון במשלוח מידע רגיש בדואר אלקטרוני, בהעברת מסמכים הכוללים חלקים פנימיים וחסויים והחלפת מפתח הצפנת המידע הקיים בשירות התשלומים וכיו"ב.

1.7 המסמך נכתב, יעודכן ויתוקף על ידי ראש תחום מתודולוגיות הגנה בסייבר – במערך ההגנה בסייבר של מערך הדיגיטל הלאומי – עבור עובדי מערך הדיגיטל הלאומי ולקוחותיו – משרדי ממשלה ויחידות סמך – המארחים את מערכותיהם ומנגישים את שירותיהם בתשתיות מערך הדיגיטל הלאומי.

2. הצפנה בתנועה

2.1 מבוא

2.1.1 מסמך מדיניות זה מתווה את מדיניות הארגון לגבי אופן היישום של הגנה על שכבות התעבורה עבור אפליקציה המשתמשת בפרוטוקול Transport Layer Security (TLS). יישום נכון של TLS יכול לספק מספר יתרונות אבטחה:

- **סודיות** - הגנה מפני קריאת תוכן תעבורת הרשת על-ידי תוקף.
- **תקינות** - הגנה מפני שינוי תוכן תעבורת הרשת על-ידי תוקף.
- **מניעת הפעלה חוזרת** - הגנה מפני הפעלה חוזרת של בקשות לשרת על-ידי תוקף.
- **אימות** - מאפשר ללקוח לאמת שהוא מחובר לשרת האמיתי (שים לב כי זהותו של הלקוח אינה מאומתת, אלא אם נעשה שימוש באישורי לקוח).

2.1.2 פרוטוקולים רבים אחרים משתמשים ב-TLS בדרכים שונות במטרה לספק הצפנה ותקינות. ההנחיות המפורטות במסמך זה מתמקדות בעיקר באופן השימוש ב-TLS לצורך הגנה על לקוחות המתחברים לאפליקציית אינטרנט באמצעות HTTPS; למרות שרוב ההנחיות הללו חלות גם על שימושים אחרים ב-TLS.

2.1.3 במערך הדיגיטל הלאומי קיימת מדיניות של HTTPS-ONLY בחוות האירוח – יש לפעול על פי מדיניות זו ולא לעשות שימוש ב-HTTP – גם כאשר הקישור המושתת על HTTP המפנה אל קישור אחר המושתת אל HTTPS. לפרטים נוספים קרא במסמך המדיניות בנושא.

2.1.4 כהחרגה לכלל בנושא HTTPS-ONLY: ניתן לעשות שימוש ב-HTTP לצורך בדיקת תקינות תעודה ברשימת ה-CRL (Certificates Revocation List) והחצנת שירות OCSP.

2.1.5 כפועל יוצא של מדיניות HTTPS-ONLY ומטרה לעמוד בדרישות תקן PCI-DSS (בכל הקשור לסביבת נתוני האשראי של הארגון) יש לעשות שימוש רק בגרסה האחרונה של פרוטוקול TLS (Transport Layer Security) - גרסה אחרונה אשר אושרה על ידי מערך ההגנה בסייבר המפורטת גם בנוהל אירוח והעלאת אתר של מערך הדיגיטל הלאומי.

2.1.6 כל שימוש בגרסה נמוכה יותר מחייב קבלת אישור להחרגה אשר יתועד על פי נוהל ניהול מערכת והוראות אבטחת המידע וההגנה בסייבר בנושא.

2.1.7 אבטחת מאגרי מידע פרטי : הצפנת המידע נועדה לתרום גם לאבטחת מאגרי המידע הפרטי הקיימים בתשתיות מערך הדיגיטל הלאומי ובכך לתרום להגנת הפרטיות של תושבי ישראל שהמידע הפרטי אודותיהם קיים במאגרי מידע אלו. על הצפנת המידע להיעשות הן ברמת מאגר המידע והן ברמה האפליקטיבית של המערכת המעבדת ומעבירה את מידע פרטי זה. בהצפנת המידע חובה לנהל נכונה גם את המפתח המשמש להצפנה – אורכו, אחסונו תוך פיצולו והחלפתו מעת לעת – ולהסדיר זאת בנוהל מאושר.

2.2 תצורת רשת

2.2.1 **יש לתמוך רק בפרוטוקולים חזקים**

2.2.1.1 לפרוטוקולים ישנים וגרסאות ישנות יש מספר רב של חולשות אבטחת מידע, ואין לעשות בהם שימוש. על אפליקציות אינטרנט לתמוך רק בגרסה אחרונה ומאושרת של TLS ולהפוך את שאר הפרוטוקולים הישנים וגרסאות קודמות ללא זמינים.

הערה: במקרים שבהם ידוע כי על שרת אינטרנט לתמוך בלקוחות מדור קודם עם דפדפנים שאינם נתמכים (כגון Internet Explorer 10), ייתכן שיהיה צורך בהפיכת גירסה לא נתמכת של TLS לזמינה על מנת לתמוך בדפדפנים אלו. יש לקבל אישור מערך ההגנה בסייבר להחרגה זו ולתעד זאת כנדרש.

2.2.2 **יש לתמוך רק בצפנים מאושרים לשימוש במערך הדיגיטל הלאומי.**

2.2.2.1 קיים מספר גדול של צפנים הנתמכים על ידי TLS ומספקים רמות אבטחה שונות. במידת האפשר, יש לאפשר עבודה רק עם צפני GCM (Galois/Counter Mode – הצפנה סימטרית). יחד עם זאת, אם נדרשת תמיכה ב-Clients מדור קודם, ייתכן שיהיה צורך בצפנים אחרים.

הערה: אין לאפשר תמיכה בפרוטוקולי CBC אשר נמצאו בהם פגיעויות והוצאו משימוש.

2.2.2.2 יש למנוע את השימוש בסוגי הצפנים הבאים:

2.2.2.2.1 צפנים ריקים (Null)

2.2.2.2.2 צפנים אנונימיים

2.2.2.2.3 חבילות צפנים אשר משרד ההגנה האמריקאי אישר בזמנו לייצוא אל מחוץ לארה"ב ונחשבות חלשות יחסית.

2.2.3 **יש להשתמש בפרמטרים חזקים של פרוטוקול דיפי-הלמן**

2.2.3.1 כאשר משתמשים בצפנים העושים שימוש בשיתוף מפתח בפרוטוקול דיפי-הלמן (מסומן במחרוזות "DHE" או "EDH" בשם הצופן) יש להשתמש בפרמטרים מאובטחים מספיק של פרוטוקול דיפי-הלמן (2048 סיביות לכל הפחות)

2.2.4 **יש להפוך דחיסה ללא זמינה**

2.2.4.1 יש להפוך את דחיסת TLS ללא זמינה על מנת להגן מפני פגיעות (הידועה בשם CRIME) העלולה לאפשר לתוקף לשחזר מידע רגיש כגון קבצי Cookie של הפעלה.

2.2.5 **יש להתקין טלאי אבטחה בספריות קריפטוגרפיות**

2.2.5.1 על מנת למנוע את נקודות התורפה בפרוטוקולי SSL ו-TLS, כאשר Heartbleed היא המפורסמת מכולן, חשוב לעדכן בהקדם האפשרי ובמסגרת הזמן אשר נקבעה על ידי מערך ההגנה בסייבר את ספריות אלו באמצעות התקנת טלאי האבטחה העדכניים ביותר.

2.2.6 **יש לבדוק את תצורת השרת**

2.2.6.1 יש לבדוק את תצורת השרת לאחר הקשחתו. לפרטים נוספים קרא בפרק 'מדריך הבדיקות של OWASP בנושא בדיקת SSL/TLS'.

2.2.6.2 ישנם מספר כלים מקוונים ולא מקוונים שבהם ניתן לעשות שימוש ע"מ לאמת במהירות את תצורת השרת. ביניהם ניתן למצוא את :

- SSL Labs Server Test
- Observatory by Mozilla

2.3 תעודות דיגיטליות

2.3.1 יש להשתמש במפתחות חזקים ולהגן עליהם

2.3.1.1 המפתח הפרטי המשמש ליצירת הצופן חייב להיות חזק מספיק למשך פרק הזמן אשר נעשה שימוש בו ובתעודה הדיגיטלית. ההמלצה היא לבחור מפתח בגודל 2048 סיבית לכל הפחות. ניתן למצוא באתר NIST מידע נוסף אודות השפעת חוזק מפתח על משך השימוש המומלץ בו.

2.3.1.2 יש להגן על המפתח הפרטי מפני גישה בלתי מורשית באמצעות הגבלת הרשאות הגישה לקבצי המערכת ובאמצעות בקרות טכניות וניהוליות.

2.3.1.3 למניעת סיכון של היחשפות המפתח הפרטי והגעתו לידיעת מי שאינו מורשה לקבלו, יש לתת עדיפות גבוהה לניהול התעודה לאורך כל מחזור חייה בתוך רכיב HSM ככל שיש רכיב כזה ברשת (לפרטים נוספים קרא בפרק נפרד במסמך זה). שימוש ב-HSM מאפשר אחסון המפתח וגם צריכתו/משיכתו על ידי רכיבי אבטחת המידע הקיימים ברשת.

2.3.1.4 בהיעדר רכיב HSM – יש לשמור על המפתח בכספת ייעודית.

2.3.1.5 בהיעדר שני רכיבים אלו – HSM וכספת – יש להגן על המפתח באמצעות הצפנה.

2.3.2 יש להשתמש באלגוריתמי Hashing קריפטוגרפים חזקים

2.3.2.1 על התעודות הדיגיטליות להיות מבוססות SHA-256 לפחות לצורך הצפנתן ולא באלגוריתמים של MD5 ו-SHA-1 הישנים יותר. אלגוריתמים ישנים אלה מאופיינים במספר חולשות קריפטוגרפיות ואינם קבילים בדפדפנים חדשים.

2.3.3 יש להשתמש בשמות מתחם (Domain) מתאימים

2.3.3.1 שם המתחם (Domain Name) של התעודה חייב להיות תואם לשם המלא של השרת המציג את התעודה.

הסבר: שם המתחם נשמר בעבר בתכונה commonName (CN) של התעודה. עם זאת, גרסאות מתקדמות של Chrome מתעלמות מתכונת ה-CN ומחייבות כי ה-FQDN יאוחסן בתכונה subjectAlternativeName (SAN). מטעמי תאימות, על ה-FQDN של התעודות להיות מאוחסן ב-CN ועל הרשימה המלאה של שמות ה-FQDN להיות מאוחסנת ב-SAN.

2.3.3.2 בנוסף, בעת יצירת התעודה הדיגיטלית, יש לקחת בחשבון את הנקודות הבאות:

- יש לשקול האם לכלול גם את מתחם המשנה (Subdomain) "www" – זאת מאחר ובמידה ולא תכלול את שם מתחם המשנה www המשתמש עלול לקבל שגיאה.

  הערה: בעולמות ה-CDN נדרשות תעודות ייחודיות המונפקות על ידי שירות ה-CDN.
- אין לכלול שמות מארחים שאינם מלאים (כגון www.gov במקום www.gov.il).
- אין לייצר תעודה דיגיטלית לכתובות IP במקום לשם מתחם.
- אין לכלול שמות מתחם פנימיים באישורים הפונים לעולם החיצון.
- אם שרת נגיש באמצעות שמות FQDN פנימיים וחיצוניים, הגדר אותו עם מספר תעודות דיגיטליות שונות.

2.3.4 יש לשקול היטב את השימוש ב-Wild Cards Certificates

2.3.4.1 השימוש בתעודות מסוג Wild Cards יכול להקל על העבודה, אך יחד עם זאת אופן עבודה זה מפר את העיקרון של ההרשאה הנמוכה ביותר, מכיוון שתעודה בודדת חלה על כל מתחמי המשנה של המתחם הראשי (כגון *.example.org). כאשר מספר מערכות חולקות תעודה כללית אחת, גוברת הסבירות שהמפתח הפרטי של התעודה ייחשף. בנוסף, הערך של תעודה כזו עולה משמעותית, מה שהופך אותה למטרה אטרקטיבית יותר לתוקפים.

2.3.4.2 הסוגיות הכרוכות בשימוש בתעודות מסוג Wide Card הן מורכבות וניתן למצוא ברשת דיונים רבים בנושא זה.

2.3.4.3 בעת הערכת הסיכונים הכרוכים בשימוש בתעודות Wild Card, יש לקחת בחשבון את הנקודות להלן:

2.3.4.3.1 השתמש בתעודות כאלו רק כאשר יש בכך צורך אמיתי ולא לשם נוחות העבודה. **הערה:** יש לקבל אישור לכך ממערך ההגנה בסייבר ולתעד ההחרגה כנדרש.

2.3.4.3.2 לעולם אל תשתמש בתעודות מסוג Wild Card עבור מערכות ברמות אמון שונות.

2.3.4.3.3 שני ממשקי VPN יכולים להשתמש בתעודת Wild Card משותפת.

2.3.4.3.4 מופעים מרובים של אפליקציית אינטרנט יכולים לחלוק תעודה דיגיטלית אחת.

2.3.4.3.5 אין לשתף תעודת Wild Card אחת לממשק VPN ושרת אינטרנט ציבורי.

2.3.4.3.6 אין לשתף תעודת Wild Card אחת על שרת אינטרנט ציבורי ושרת פנימי.

2.3.4.3.7 יש לעשות שימוש בשרת Reverse Proxy המבצע פתיחת הצפנת TLS במקום מערכות ההפעלה, כך שהמפתח הפרטי של תעודת ה-Wild Card יימצא רק במערכת אחת. כל בקשה להחרגה צריכה להיות מאושרת על ידי מערך ההגנה בסייבר ולהיות מתועדת כנדרש.

2.3.4.3.8 יש לתחזק רשימה של כל המערכות החולקות תעודת Wild Card כדי לאפשר לעדכן את כולן באם תוקף תעודת ה-Wild Card פג או שיש להחליפה עקב חשיפה.

2.3.4.3.9 יש להגביל את ההיקף של השימוש בתעודות Wild Card על-ידי הנפקתן למתחם משנה מסוים (כגון *.foo.example.org) או למתחם נפרד.

2.3.4.3.10 יש לייבא את תעודת ה- Wild Card לשרת ללא אפשרות ייצוא.

2.3.5 יש להשתמש ב-CA מוכר ומקובל על משתמשי האפליקציה

2.3.5.1 על מנת להעלות את אמינות האפליקציות המפותחות בארגון, על האישורים להיחתם על-ידי CA (גורם מאשר) מהימן ומקובל. עבור אפליקציות הפונות לאינטרנט, גורם מאשר (CA) זה אמורה להיות אחד הגורמים המאשרים המוכרים בה מערכות הפעלה (ובכללם טלפונים חכמים) ודפדפנים נפוצים נותנים אמון באופן אוטומטי.

2.3.5.2 עדיף להימנע מלעשות שימוש ב-CA חינמי כדוגמת LetsEncrypt אשר מספקת אישורי SSL חינמיים לאתרים ושירותי רשת, שבהם מרבית הדפדפנים העיקריים נותנים אמון. יש לשקול בזהירות את יתרונות

השימוש בשירות חינמי זה עבור שירותים ממשלתיים ומומלץ להשתמש ב-CA אמין, מהימן והמקובל על מגוון רחב יותר של מערכות ומשתמשים.

2.3.5.3 עבור אפליקציות פנימיות, ניתן להשתמש בגורם מאשר (CA) פנימית. משמעות הדבר היא שה-FQDN של האישור לא ייחשף (לא לגורם מאשר חיצוני ולא באופן ציבורי ברשימות CA מוכרים בעולם).

2.3.6 **לבדיקה על ידי תחום יישומי אבטחת מידע: יש להשתמש ברשומות CAA כדי להגביל את גורמים מאשרים היכולים להנפיק אישורים**

2.3.6.1 ניתן להשתמש ברשומות CAA (מתן הרשאות ל-CA) של DNS כדי להגדיר אילו CA-ים רשאים להנפיק אישורים למתחם (Domain). הרשומות מכילות רשימה של CA-ים, וכל CA שאינו כלול ברשימה זו נדרש לסרב להנפיק אישור ל-Domain. כך ניתן למנוע מתוקף מלקבל תעודה אסורה באמצעות CA מהימן פחות. כאשר מחילים רשומות אלה על כל תחומי המשנה (Sub-Domain), הדבר נותן יתרון למנהל המערכת על-ידי הגבלת ה-CA-ים שבהם עובדי הפרויקט יכולים להשתמש, ולמנוע מהם מלהשיג אישורים כלליים ממקורות בלתי-מורשים.

2.3.7 **יש לספק תמיד את כל האישורים הדרושים**

2.3.7.1 בשרשרת גורמים מאשרים (Certificate Chain of Trust) – יש גורם מאשר ראשי (עוגן האמון) וגורם מאשר מתווך. במקרים רבים האישור אינו חתום ישרות על ידי גורם מאשר (CA) ראשי אלא על ידי גורם מאשר מתווך.

2.3.7.2 חובה לזכור כי הדפדפן של המשתמש בוחן את האישור ששימש לחתימה על מנת לאמת את מהימנות האישור ומשווה אותו לרשימת הגורמים המאשרים שבהם המערכת של המשתמש נותנת אמון.

2.3.7.3 אם המשתמש אינו מכיר גורם מאשר (CA) המתווך ולא נותן בו אמון, אזי אימות האישור ייכשל, גם אם המשתמש נותן אמון בגורם המאשר

(CA) הראשי, מאחר שאין באפשרותו לבסס שרשרת אמון בין האישור לבין הגורם המאשר (CA) הראשי.

2.3.7.4 לכן, יש לצרף גם את האישורים של ה-CA-ים הגורמים המאשרים המתווכים של התעודה החתומה – ולא רק את האישור של ה-CA הראשי.

2.3.8 יש לשקול את השימוש באישורי אימות מורחבים

2.3.8.1 נטען כי אישורי אימות מורחב (EV) מספקים רמה גבוהה יותר של אימות הישות, שכן הם מבצעים בדיקות כי המבקש הוא ישות משפטית לגיטימית, ולא רק מאמתים את אישורי הבעלות על שם המתחם. זהו ההבדל בין *"אתר זה אכן מנוהל על-ידי חברה X"* לעומת *"מתחם זה הינו אכן example.org"*.

2.3.8.2 אישורים אלה הוצגו בעבר באופן שונה בדפדפן, לרוב על-ידי הצגת שם החברה או סמל/רקע ירוקים בשורת הכתובת. החל משנת 2019, דפדפני Chrome ו-Firefox הודיעו על הסרת מחוונים אלה, מאחר שלדעתם אישורי EV אינם מספקים כל יתרון.

2.3.8.3 אין מניעה להשתמש באישורי EV. עם זאת, מכיוון שאישורים אלה יקרים משמעותית לעומת אישורי מתחם מאומתים, ובמיוחד כארגון ממשלתי, אין ערך לתעודה חתומה עם EV.

2.4 אפליקציה

2.4.1 יש להשתמש ב-TLS לכל הדפים

2.4.1.1 יש להשתמש ב-TLS לכל הדפים, ולא רק לרכיבים שנחשבים לרגישים כגון דף הכניסה. אם ישנם דפים שאינם אוכפים את השימוש ב-TLS, הם עשויים לתת לתוקף הזדמנות לרחרח אחר מידע רגיש כגון אסימוני הפעלה (Tokens), או להחדיר קוד JavaScript זדוני לתגובות לצורך ביצוע התקפות אחרות המתמקדות במשתמש.

2.4.1.2 מערך הדיגיטל הלאומי הפיץ את מסמך מדיניות HTTPS Only במטרה למנוע מניפולציות באתרים המאפשרים עדיין גישה ב-HTTP.

2.4.1.3 במידה ואפליקציה עושה מסיבה כזו או אחרת שימוש ב-HTTP, ניתן לאפשר לשרת האינטרנט להאזין לחיבורי HTTP לא מוצפנים ביציאה

80, ואז לנתב אותם מיד עם ניתוב קבוע מחדש (HTTP 301) בתעבורת HTTPS – ובכך להתגבר על הבעיה הנוצרת כאשר משתמשים מקלידים ידנית את שם מתחם.

2.4.1.4 על מנת למנוע מהמשתמשים מלגשת בעתיד לאתר באמצעות HTTP יש להוסיף בדפי האתר את הכותרת (Header) HTTP Strict Transport Security (HSTS).

2.4.2 אין לשלב תכני TLS עם תכנים שאינם תכני TLS

2.4.2.1 דף הזמין באמצעות TLS לא אמור לכלול קבצי משאבים (כגון JavaScript או CSS) אשר נטענים באמצעות HTTP לא מוצפן. משאבים לא מוצפנים אלה עלולים לאפשר לתוקף לחקור אחר קבצי Cookie באתר או להחדיר קוד זדוני לדף. דפדפנים מתקדמים יחסמו גם ניסיונות לטעון לדפים מאובטחים כל תוכן פעיל באמצעות HTTP – או יקפיצו התראה לגולש על בעיה באמינות האתר.

2.4.3 יש להשתמש בתכונת Secure עבור קבצי ה-Cookie.

2.4.3.1 יש לסמן את כל קבצי ה-Cookie בתכונה "מאובטח", המורה לדפדפן לשלוח אותם רק באמצעות חיבורי HTTPS מוצפנים, על מנת למנוע ציתות בחיבור HTTP לא מוצפן. לדרישה זו יש חשיבות גם אם האתר לא מאזין ב-HTTP (יציאה 80), שכן תוקף שמבצע התקפה פעילה מסוג 'אדם באמצע' (MiTM – Man in The Middle) יכול להציג למשתמש שרת אינטרנט מזויף בפורט 80 כדי לגנוב את קובץ ה-Cookie שלו והמידע אשר קיים בו ולהמשיך התחזות ל-Client.

2.4.4 יש למנוע אחסון מידע רגיש בזיכרון ה-Cache (מטמון).

2.4.4.1 למרות שפרוטוקול TLS מספק הגנה על נתונים במעבר, הוא אינו מספק שום הגנה על נתונים ברגע שהם מתקבלים או מאוחסנים במערכת המבקשת. מידע זה עלול להיות מאוחסן בזיכרון המטמון של הדפדפן של המשתמש ולהגנב משם. אפשרות אחרת היא שיאוחסן בשרת Proxy הנמצא בתווך ומוגדר לביצוע פענוח של TLS.

2.4.4.2 כאשר מידע רגיש מוחזר ב-Response, יש להשתמש בכותרות HTTP כדי להורות לדפדפן ולשרתי Proxy הנמצאים בתווך לא לאחסן בזכרון

המטמון (ה-Cache) את המידע, על מנת למנוע את אחסונו או איחזורו למשתמשים אחרים. ניתן להשיג זאת על-ידי הגדרת כותרות ה-HTTP הבאות ב-Response:

```
Cache-Control: no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: 0
```

2.4.5 הקפד על תעבורה מאובטחת בפרוטוקול HTTP

2.4.5.1 הקפדה על תעבורה מאובטחת בפרוטוקול HTTP (HSTS) מורה לדפדפן של המשתמש לבקש תמיד את האתר באמצעות HTTPS, וגם מונעת מהמשתמש מלהתעלם מאזהרת תעודות שנחתמו על ידי רשות חתימה לא מוכרת. עיין ב הוראות בנושא הקפדה על תעבורה מאובטחת בפרוטוקול HTTP לצורך קבלת מידע נוסף על יישום HSTS.

2.4.6 יש להשתמש בהצמדת מפתח ציבורי

2.4.6.1 עשה שימוש ב-SSL Pinning או DNS SEC (TBD).

2.4.6.2 הצמדת מפתח ציבורי של HTTP (HPKP) משמשת לצימוד של אישור ספציפי לשם מתחם, על מנת למנוע מתוקף מלבצע התקפה מסוג MiTM באמצעות אישור שונה (אך מהימן). עיין במאמר 'אישור והצמדת מפתח ציבורי' לקבלת מידע נוסף על יישום HPKP.

2.4.6.3 אין לעשות שימוש ב-HPKP. השימוש בו מהווה פתח לסיכונים, הוא שנוי במחלוקת ונתמך רק על ידי חלק מהדפדפנים.

2.4.7 יש לשקול את השימוש בתעודות צד לקוח

2.4.7.1 באתרים בהם נדרש זיהוי של הלקוח יש לעשות שימוש בתעודות צד לקוח.

2.4.7.2 בתצורה טיפוסית נעשה שימוש ב-TLS ביחד עם תעודה בשרת כדי שהלקוח יוכל לאמת את זהות השרת, ולספק חיבור מוצפן ביניהם. עם זאת, בגישה זו יש שתי חולשות עיקריות:

2.4.7.2.1 לשרת אין שום מנגנון לאימות זהות הלקוח

2.4.7.2.2 יש לתוקף אפשרות ליירט את החיבור על-ידי השגת תעודה חוקית עבור שם המתחם (DNS).

2.4.7.3 ארגונים משתמשים לעתים קרובות בתעודות אלו כדי לבצע בדיקה של תעבורת TLS על-ידי התקנת תעודה אמינה של גורם מאשר (CA) במערכות צד לקוח שלהם.

2.4.7.4 תעודות לקוח נותנות מענה לשתי סוגיות אלה על-ידי דרישה מהלקוח להוכיח את זהותו לשרת באמצעות תעודה משלו. מנגנון זה מאפשר לא רק אימות חזק של זהות הלקוח, אלא גם מונע מגורם ביניים מלבצע פענוח TLS, גם אם יש ברשותו תעודה אמינה של גורם מאשר (CA) במערכת הלקוח.

2.4.7.5 לא נעשה שימוש רב בתעודות לקוח במערכות ציבוריות עקב מספר קשיים:

2.4.7.5.1 הנפקה וניהול של תעודות לקוח מהווה תקורת ניהול משמעותית.

2.4.7.5.2 משתמשים שאינם משתמשים טכניים עשויים להתקשות בהתקנת תעודות לקוח.

2.4.7.5.3 במקרה בו תעודת לקוח אחת משמשת לתעבורה מול ארגונים רבים, חסימה של תעודה אחת תפגע בכלל השירותים המזוהים באמצעותה.

2.4.7.6 נדרש להשתמש בתעודת צד לקוח עבור אפליקציות או ממשקי API קריטיים – mTLS - במיוחד כאשר ישנו מספר קטן של משתמשים והם חלק מאותו ארגון.

2.5 יישום בנהלים

2.5.1 יישום על ידי נהלים עבודה פנימיים

2.5.1.1 שיטות ההצפנה בתנועה – המותרות והאסורות – מפורטות בנוהל הנפקת תעודות SSL ונהלים נוספים העוסקים בשימוש בשיטות צופן המאושרים לשימוש במערך הדיגיטל הלאומי.

2.6 מאמרים קשורים

- OWASP - קובץ הוראות בנושא מחרוזת צפני TLS
- OWASP - בדיקת SSL-TLS ו-OWASP מדריך לקריפטוגרפיה
- OWASP - תקן לאימות אבטחת אפליקציות (ASVS) - דרישות לאימות אבטחת תקשורת (V10)
- Mozilla - תצורות מומלצות של Mozilla
- NIST - SP 800-52 Rev. 1 הנחיות לבחירה, להגדרת תצורה ולשימוש ביישומים של Transport Layer Security (TLS)
- NIST SP 800-57 - NIST המלצה לניהול מפתחות, מהדורה 3, טיוטה ציבורית
- NIST - SP 800-95 מדריך לשירותי אינטרנט מאובטחים
- IETF - RFC 5280 תקן X.509 של אישור תשתית מפתח ציבורי ופרופיל של רשימת אישורים מבוטלים (CRL) לשימוש באינטרנט
- IETF - RFC 2246 פרוטוקול Transport Layer Security (TLS) גרסה 1.0 (ינו' 1999)
- IETF - RFC 4346 פרוטוקול Transport Layer Security (TLS) גרסה 1.1 (אפר' 2006)
- IETF - RFC 5246 פרוטוקול Transport Layer Security (TLS) גרסה 1.2 (אוג' 2008)
- Bettercrypto - הקשחת הצפנה יישומית: כיצד לאבטח את הגדרות ההצפנה במרבית השירותים הנפוצים)

3. הצפנה במנוחה

3.1 מבוא

3.1.1 פרק זה מפרט את קווי מדיניות ההצפנה של הארגון בעת יישום פתרונות להגנה על נתונים במנוחה.

3.2 החלטה ארכיטקטונית

3.2.1 יש לקבל החלטה ארכיטקטונית כדי לקבוע את השיטה המתאימה להגנה על נתונים במנוחה. ישנם סוגים רבים של מוצרים, שיטות ומנגנונים לאחסון של נתונים מוצפנים. מסמך הנחיות זה יתמקד רק בהנחיות ברמה נמוכה עבור מפתחים וארכיטקטים המיישמים פתרונות קריפטוגרפיים. במסמך זה אין התייחסות לפתרונות של ספקים ספציפיים, ולא התייחסות לתכנון האלגוריתמים הקריפטוגרפיים.

3.2.2 שיטות העבודה הכלליות ואורך המפתח המינימלי הנדרש ייקבעו בהתאם לצורך ואופי היישום. להלן פירוט שיטות העבודה האפשריות:

- שיתוף מפתח: פרוטוקול דיפי-הלמן עם 2048 סיביות לפחות
- תקינות ההודעה: HMAC-SHA2
- קוד Hash של הודעה: SHA2 של 256 סיביות
- הצפנה אסימטרית (שליפת המידע המוצפן על ידי משתמשים רבים): RSA של 2048 סיביות
- הצפנה סימטרית (יעשה בה שימוש להצפנה ופענוח על ידי רכיב אחד אשר שומר את המידע וגם שולף אותו – כדוגמת BitLocker או הצפנת דיסק ברמת המערכת על ידי השרת): AES של 128 סיביות
- אלגוריתמי Hashing של סיסמאות: Argon2, PBKDF2, Scrypt, Bcrypt

3.3 קביעת אופן הצפנת הנתונים

3.3.1 **תכנון אחסון נתונים מוצפנים :**

3.3.1.1 על הארגון לבחון באופן מדוקדק את כל הפרוטוקולים והאלגוריתמים לאימות ולתקשורת מאובטחת.

3.3.1.2 יש לוודא כי השימוש בתעודות בדיגיטליות נעשה במתחמים שעבורם אכן מיועדות התעודות.

3.3.1.3 יש להימנע משימוש בתעודות דיגיטליות כלליות (Wild Card), אלא אם כן יש בכך צורך עסקי.

3.3.1.4 יש להקפיד על שימוש במנגנוני הצפנה העומדים בתקנים תוך הקפדה על אבטחת הרשת, אבטחת האלגוריתמים, וניהול מפתחות ההצפנה בהתאם לתקופת השימוש המותרת בהם.

3.3.2 הימנעות מאגירת מידע רגיש ללא צורך מוגדר

3.3.2.1 אגירת ***מידע רגיש מבלי שיש לך צורך עסקי – הן אגירה בכמות המידע והן במשך האגירה - מגדילה את הסיכון לחשיפת מידע רגיש רב במקרה של אירוע אבטחת מידע. יש להימנע מאגירת מידע רגיש כגון נתוני אשראי או נתוני מידע פרטי שלא לצורך או מעבר לזמן הנדרש. לפרטים נוספים יש לפנות למדיניות הגיבויים, מדיניות הגנת הפרטיות ומדיניות הפיתוח (ובכללה נספח לפיתוח מערכות בסביבת נתוני האשראי).***

3.3.3 הצפנה מאומתת חזקה ומאושרת

3.3.3.1 יש להשתמש בהצפנה מאומתת חזקה ומאושרת כגון CCM או GCM – זוהי הצפנה המבוססת על אלגוריתם AES.

3.3.3.2 ***שימוש במנגנוני הצפנה מאושרים חזקים***

- יש להשתמש בהצפנה מאושרת וחזקה. אין להטמיע במערכת אלגוריתם הצפנה אשר אינו תקני, אינו מקובל, ישן או אינו מאושר לשימוש.
- יש להשתמש רק באלגוריתמים מקובלים ומאושרים כגון AES, הצפנה של מפתח ציבורי של RSA ו- SHA-256 ואילך לצורך ביצוע Hashing. אין להשתמש באלגוריתמים חלשים כגון MD5 או SHA1. יש להימנע מביצוע Hashing לצורך אחסון סיסמאות. במקום זאת, יש להשתמש ב-Argon2, ב-PBKDF2, ב-bcrypt או

ב-scrypt. יש לשים לב שבמשך הזמן אלגוריתמים קריפטוגרפיים מאבדים מחוזקם ויש לבדוק את תקפותם.

- יש לוודא כי בפיתוח היישום היה מעורב גורם הבקיא בצרכי הארגון ועולם הקריפטוגרפיה.
- באופן כללי, אין לעשות שימוש במנגנוני הצפנה ישנים ו/או חלשים.
- בהגנה על מפתחות יש לעשות שימוש בסיסמה חזקה מספיק בהתאם לחשיבות המפתחות שעליהם היא מגנה.

3.3.3.3 *מחוללי מספרים פסאודו-אקראיים קריפטוגרפיים (CSPRNG)*

- יש להשתמש במחולל מספרים פסאודו-אקראיים קריפטוגרפי (CSPRNG) הזמינים בשפת התכנות בה נעשה פיתוח המערכת.
- לצורך יצירת מפתחות ממקורות פסאודו-אקראיים, לא מומלץ להשתמש בפונקציות אקראיות רגילות של מערכת ההפעלה .
- ניתן להשתמש במפתחות ממקורות פסאודו-אקראיים לשימושים במערכות ורכיבים לא-רגישים.
- אין להשתמש במחוללי מספרים פסאודו-אקראיים (PRNG) עבור מערכות ושימושים רגישים מכיוון שהם אינם מאובטחים ברמה גבוהה וקיים סיכון בשימוש בהם. המספרים שנבחרים אינם אקראיים לחלוטין מכיוון שנעשה שימוש באלגוריתם מתמטי לבחירתם. יחד עם זאת, הם אקראיים מספיק למטרות מעשיות ושימושים בלתי רגישים.
- מחוללי PRNG משמשים בעיקר ליצירת מספרים אקראיים לכאורה בהם ניתן לעשות שימוש במקרים כדוגמת מידול וסימולציה בתוכנה, ואילו מחוללי CSPRNG מיועדים ליצירת מספרים אקראיים לכאורה העמידים בפני חיזוי ובהם ניתן לעשות שימוש, לדוגמא, לאבטחת סיסמאות אקראיות, אסימוני CSRF, מזהי הפעלה וכו'.
- השימוש במחוללי CSPRNG יבטיח, ככל הניתן, כי כל המספרים האקראיים, במיוחד אלה המשמשים לפרמטרים קריפטוגרפיים (מפתחות, וקטורי אתחול - IV, תגיות MAC), שמות קבצים

אקראיים, מזהי GUID אקראיים ומחרוזות אקראיות נוצרים בצורה חזקה מבחינה קריפטוגרפית.

- ודא כי נעשה שימוש באלגוריתמים אקראיים עם אנטרופיה (חוסר אפשרות לחיזוי) מספקת.

3.3.3.4 ***הצפנה מאומתת:***

- הצפנה מאומתת מספקת סודיות, תקינות, ומקוריות (CIA), בעוד שהצפנה לבדה מספקת רק סודיות. יש לשלב תמיד את ההצפנה עם הגנה על תקינות ההודעה ומקוריות ההודעה. אחרת, הטקסט המוצפן עלול להיות חשוף למניפולציה, במיוחד אם הוא מועבר בערוצים לא בטוחים (כגון בכתובת URK או בקובץ Cookie).
- מומלץ להשתמש בשני מפתחות שונים עבור שתי פעולות נפרדות אלו.
- שים לב כי הגנה ע לתקינות ומקוריות עדיפה על פני הגנה על תקינות בלבד. במילים אחרות, אימות כגון HMAC-SHA256 או HMAC-SHA512 עדיף על SHA-256 או SHA-512.
- הצפנת דיסק היא מקרה מיוחד של נתונים במנוחה לדוגמה, מערכת קבצים מוצפנת בכונן דיסק קשיח. יש לנסות ולהגן על תוכן הדיסק הקשיח של - המערכת או עמדת העבודה – באמצעות יישום הצפנת דיסק מומלצת.

3.3.4 איחסון ערך Hash וביצוע Salt לסיסמאות

3.3.4.1 יש להשתמש בשיטות מורכבות (כגון Argon2, ב-PBKDF2, ב-bcrypt או ב-scrypt) הן מ-Hash והן מ-Salt לצורך אחסון סיסמאות במטרה להגן עליהן מפני פיצוח באמצעות טבלאות Hash ידועות. למידע נוסף על אחסון סיסמאות, עיין בקובץ ההוראות לאחסון סיסמאות.

3.3.5 הגנה קריפטוגרפית מאובטחת כאשר בקרות הגישה נכשלות

3.3.5.1 הצפנה תגן על המידע מפני חשיפה גם כאשר בקרות הגישה האחרות יכשלו ולא ימנעו גישה אסורה. בקרות הגישה (שמות משתמש, סיסמאות, הרשאות וכו') הן שכבת הגנה אחת. הצפנת המידע במנוחה מוסיפה שכבת הגנה נוספת אשר תמשיך להגן על הנתונים גם אם היתה גישה

אסורה אל מסד הנתונים עצמו ברמת האחסון. לכן, חשיבות חוזקו של מפתח ההצפנה, אופן אחסונו והגישה אליו הינו קריטי לשמירה על נתוני הארגון.

3.3.6 כלל - ודא שכל מפתח סודי מוגן מגישה בלתי מורשית

3.3.6.1 *כלל - הגדר מחזור חיים של מפתח*

- מחזור חיי המפתח מפרט את המצבים השונים שמפתח יעבור במהלך חייו. מחזור החיים יציין מתי כבר לא צריך להשתמש במפתח לצורך הצפנה, מתי כבר לא צריך להשתמש במפתח לצורך פענוח (מקרים אלה אינם חופפים בהכרח), אם יש להמיר מחדש את הנתונים עם הצגת מפתח חדש ומתי יש להוציא מפתח משימוש.

3.3.6.2 *כלל - אחסן מפתחות לא מוצפנים הרחק מהנתונים המוצפנים*

- אם המפתחות מאוחסנים יחד עם הנתונים, אזי כל חשיפה של הנתונים לסכנה מהווה גם חשיפה של המפתחות. לעולם אין לאחסן מפתחות לא מוצפנים באותה תשתית בה מאוחסנים הנתונים.

3.3.6.3 *כלל - השתמש במפתחות עצמאיים כשנדרשים מספר מפתחות*

- ודא שהמפתחות הם בלתי תלויים. כלומר, אל תבחר במפתח שני הקשור בקלות למפתחות הראשונים (או לכל מפתח קודם).

3.3.6.4 *כלל - הגן על המפתחות במאגר מפתחות*

- יש לאחסן בכל עת את המפתחות במאגר מפתחות מוגן. יש לודא שקיים פער בין אפשרויות התקיפה של הנתונים לבין אפשרויות התקיפה של המפתחות.
- משמעות הכלל היא שאין לאחסן מפתחות באפליקציה או בשרת לו יש גישה לרשת האינטרנט (בהנחה שתוקפי אפליקציות הם חלק ממודל האיומים הרלוונטי).

3.3.6.5 ***כלל - תעד נהלים קונקרטיים לניהול מפתחות לאורך מחזור החיים***

- יש להסדיר בנהלים את תהליכי העבודה עם מפתחות ולהדריך כהלכה את האחראים על המפתחות.
- **הערה:** מדיניות פיתוח מערכות מאובטחות מגדירה את הכללים לניהול מפתחות לאורך מחזור המערכת וביניהם איסור שמירה ע"י השרת עצמו, ע"ג שרת FTP, במערכת ההלבנה ובמקומות אחסון אחרים הנגישים או עלולים להיות נגישים בפני גורמים בלתי מורשים.

3.3.6.6 ***כלל – הסדר בנוהל את שינוי אלגוריתמים ומפתחות בעת הצורך***

- במקרה של חשיפת מפתחות או שיפוג תוקפו של גורם מאשר חיצוני, יהיה צורך בהחלפת מפתחות. יש להסדיר במסמך מדיניות ובנוהל את תהליך זה ולחייב את מנהלי המערכות, המוצרים והשירותים להחליף בתנאים מסוימים את המפתחות ולהצפין מחדש את הנתונים.
- יש להיערך למתן מענה מהיר לצורך כזה כאשר הדבר נדרש. קשירה של גרסת מפתח וגרסת אלגוריתם הצפנה ביחד עם הנתונים המוצפנים היא אפשרות שימושית ופרואקטיבית.
- למשל, הוספת מחרוזת התחלתית פשוטה, כגון "{1,1}..."., לפני הנתונים המוצפנים יכולה לציין גרסת אלגוריתם 1 וגרסת מפתח 1. פעולה זו מאפשרת שינוי "מקוון" לאלגוריתם ההצפנה ולמפתח מבלי להצפין מחדש בבת אחת את כל הנתונים הקיימים.

3.3.6.7 ***כלל - תעד נהלים קונקרטיים לטיפול בחשיפה של מפתחות לסכנה***

- יש לודא שלצוותי התפעול יש את המידע הדרוש להם ושמידע זה זמין בקלות כאשר יש לבצע החלפה של מפתחות הצפנה. החלפה של מפתחות לא אמורה לחייב שינויים בקוד המקור או בתהליכי יישום מסובכים, שכן ביצוע פעולת כאלו במהלך אירוע יכביד עוד יותר על הצוות ועלול לגרום לתקלה.

3.3.6.8 ***כלל - הגבל את כמות הנתונים המוצפנים באמצעות מפתח אחד***

- אם כמות הנתונים המוצפנים גדלה מעבר ל**סף מסוים**, יש להשתמש במפתח חדש. **סף מסוים** זה משתנה בהתאם

לאלגוריתם ההצפנה בשימוש, אך בדרך כלל הוא 2^{35} בתים (34 גיגה-בתים בקירוב) עבור צפני בלוקים של 64 סיביות (DES, DES3, Blowfish, RC5, ...) ו-2^{68} בתים (295,147,905 טרה-בתים בקירוב) עבור צפני בלוקים של 128 סיביות (AES, TwoFish, Serpent). סביר להניח שסף זה לא יושג בעת הצפנה באמצעות צופן חדשני, אך יש לקחת אותו בחשבון בעת הערכת אלגוריתמים ונהלי החלפה.

3.3.7 כלל - פעל בהתאם לתקנות הרלוונטיות לשימוש בקריפטוגרפיה

3.3.7.1 *כלל - עליך להגן על נתוני בעלי הכרטיסים בהתאם לדרישה 3 של תקן PCI DSS*

- תקן אבטחת הנתונים (DSS) של בתעשיית כרטיסי האשראי (PCI) פותח כדי לעודד ולשפר את אבטחת המידע של מחזיקי כרטיסי האשראי ולסייע באימוץ נרחב של אמצעי אבטחת מידע הנפוצים בעולם. התקן הוצג בשנת 2005 והוא החליף תקני אבטחת מידע פרטניים של חברות כרטיסי האשראי ויזה, מאסטרקארד, אמריקן אקספרס, JCB ודיינרס.
- יישום תקן PCI DSS במהדורתו האחרונה – כמתחייב במערך הדיגיטל הלאומי - נותן מענה לאחסון מאובטח של נתוני כרטיסי אשראי. דרישה זו מספקת מענה לכמה היבטים של אחסון מאובטח של נתוני האשראי. במסמך מדיניות זה אנו כוללים אחסון מוצפן בהתאם לדרישות 3.4, 3.5 ו- 3.6 כפי שתוכל לראות להלן:

3.4 PAN (מספר חשבון ראשי)

יש לעבוד עם PAN (מספר חשבון ראשי), בדרך כזו אשר לכל הפחות תביא לכך שלא יהיה קריא במקום שבו הוא מאוחסן

ניתן לעמוד בדרישה 3.4 על-ידי יישום של כל אחד מארבעת סוגי האחסון המאובטח המתוארים בתקן הכולל קידוד ונתוני Hashing. לרוב, שתי הגישות הללו יהיו הבחירות הפופולריות ביותר מרשימת האפשרויות. התקן אינו מתייחס לאלגוריתמים ספציפיים אך הוא מחייב שימוש ב**קריפטוגרפיה חזקה**. מסמך מילון המונחים של מועצת PCI העולמי מגדיר **קריפטוגרפיה חזקה** כ:

קריפטוגרפיה המבוססת על אלגוריתמים שנבדקו ואשר מקובלים בענף, יחד עם אורכי מפתח חזקים ושיטות עבודה מתאימות לניהול מפתחות. קריפטוגרפיה היא שיטה להגנה על נתונים וכוללת גם הצפנה (הפיכה) ו-Hashing (שאינו הפיך, או "כיוון אחד"). SHA-1 הוא דוגמה לאלגוריתם Hashing שנבדק ומקובל בענף. דוגמאות לתקנים ולאלגוריתמים להצפנה שנבדקו ומקובלים בענף כוללים AES (128 סיביות ואילך), TDES (מפתחות באורך כפול לפחות), RSA (1024 סיביות ואילך), ECC (160 סיביות ואילך), ו-ElGamal (1024 סיביות ואילך).

אם יישמת את הכלל השני בקובץ הנחיות זה, עליך ליישם אלגוריתם קריפטוגרפי חזק תואם העומד בדרישה 3.4 של תקן PCI DSS או חזק יותר. עליך לוודא שאתה יודע לזהות את כל המקומות שבהם מאוחסנים נתוני כרטיסים, כולל יומני רישום, ולהחיל עליהם את רמת ההגנה המתאימה. הדבר יכול לנוע בין הצפנת הנתונים להחלפת מספר הכרטיס ביומני הרישום.

ניתן לעמוד בדרישה זו גם על-ידי יישום הצפנת דיסק במקום הצפנה ברמת הקובץ או העמודה. הדרישות **לקריפטוגרפיה חזקה** זהות גם להצפנת דיסק ולמדיית גיבוי. אין לאחסן את נתוני הכרטיסים בצורה גלויה. על-ידי ביצוע ההנחיות בקובץ הנחיות זה תוכל לאחסן את הנתונים שלך בצורה מאובטחת באופן התואם לדרישה 3.4 של תקן PCI DSS.

3.5 מפתחות

יש להגן על כל המפתחות המשמשים לאבטחת נתוני מחזיקי כרטיסי האשראי מפני חשיפה ושימוש לרעה

כפי ששם הדרישה לעיל מרמז, יש לאחסן את מפתחות ההצפנה באופן מאובטח. דבר זה יחייב יישום של בקרת גישה חזקה, ביקורת ורישום של המפתחות שלך. יש לאחסן את המפתחות במקום שהוא גם מאובטח וגם "רחוק" מהנתונים המוצפנים. משמעות הדבר היא כי אין לאחסן נתוני מפתחות בשרתי אינטרנט, בשרתי מסדי נתונים וכו'. יש לפעול על פי הנוהל הפנימי של הארגון המסדיר את החלפת המפתחות.

יש לצמצם את קבוצת מורשי הגישה למפתחות. על קבוצת משתמשים להיות מורכבת ממשתמשים מהימנים ומיומנים מאוד לביצוע התפקידים של 'נאמן מפתח'. כמובן שיידרשו חשבונות מערכת/שירות לצורך גישה לנתוני המפתחות לצורך הצפנה/פענוח של הנתונים.

אין לאחסן את המפתחות עצמם בצורה גלויה ויש להצפין אותם באמצעות KEK (מפתח להצפנת מפתחות). אין לאחסן את ה-KEK באותו מיקום שבו מאוחסנים מפתחות ההצפנה שאותם הוא מצפין.

3.6 תהליכים ונהלים

יש להסדיר, לתעד וליישם באופן מלא את כל התהליכים והנהלים לניהול מפתחות עבור מפתחות קריפטוגרפיים המשמשים להצפנה של נתוני מחזיקי כרטיסי האשראי.

דרישה 3.6 מחייבת כי כל תהליכי ניהול המפתחות בארגון עונים על דרישות תקן PCI ויכללו 8 שלבים ספציפיים במחזור החיים:

3.6.1 יצירת מפתחות קריפטוגרפיים חזקים

כמפורט בתחילת מסמך זה, יש להשתמש באלגוריתמים המאפשרים רמות גבוהות של אבטחת מידע. יש ליצור מפתחות חזקים כדי למנוע פגיעה באבטחת הנתונים העלולה להגרם במקרה ונשתמש במפתחות קריפטוגרפיים חלשים. מפתח חזק נוצר כאשר אורכו עונה על לדרישות אבטחת הנתונים שלך והעומד בדרישות של תקן PCI DSS. אורך המפתח בלבד אינו מהווה מדד לחוזק של מפתח. הנתונים המשמשים ליצירת המפתח חייבים להיות אקראיים מספיק (לעתים קרובות, "מספיק" נקבע לפי דרישות אבטחת המידע שלך) והאנטרופיה של נתוני המפתחות עצמם חייבת להיות גבוהה.

3.6.2 הפצה מאובטחת של מפתחות קריפטוגרפיים

השיטה המשמשת להפצת מפתחות חייבת להיות מאובטחת כדי למנוע גניבת מפתחות בהעברתם מנקודה לנקודה. השימוש בפרוטוקול כדוגמת דיפי-הלמן יכול לסייע באבטחה של הפצת המפתחות והשימוש בתעבורה מאובטחת כדוגמת TLS ו-SSHv2 יכול אף הוא לאבטח את המפתחות במעבר. אין להשתמש בפרוטוקולים ישנים כגון SSLv3.

3.6.3 אחסון מאובטח של מפתחות קריפטוגרפיים

לגבי אחסון מאובטח של מפתחות הצפנה, כולל מפתחות KEK, ר' תיאור דרישה 3.5 (ראה לעיל).

3.6.4 החלפה עיתית של המפתחות הקריפטוגרפיים

תקן ה-PCI DSS מחייב להחליף פעם בשנה לפחות את המפתחות המשמשים להצפנה. על תהליך החלפת המפתחות להסיר מפתח ישן מתהליך ההצפנה/פענוח ולהחליף אותו במפתח חדש. יש להצפין את כל הנתונים החדשים שנכנסים למערכת באמצעות המפתח החדש. למרות שמומלץ להצפין מחדש את הנתונים הקיימים באמצעות המפתח החדש לפחות פעם בשנה או בשלוש שנים בהתאם לנוהל הפנימי של הארגון העוסק בהצפנה מחדש של נתונים והחלפת מפתח, תקן PCI DSS אינו מחייב זאת במפורש.

3.6.5 **הוצאה משימוש או החלפה של מפתחות בהתאם לצורך כאשר תקינות המפתח נחלשה או שיש חשש לחשיפה של מפתחות לסכנה**

נוהל החלפת מפתחות בשירות התשלומים מסדיר את תהליכי ניהול המפתחות ובכלל זה מפתחות המאוחסנים בארכיון, מפתחות שהוצאו משימוש או מפתחות אשר קיים חשש שנחשפו. על הנוהל להסדיר את התהליכים בכפוף לדרישות 3.6.2, 3.6.3 ו-3.6.4 יסדירו גם תהליכים מאובטחים לאחסון ולהחלפה של מפתחות אלה.

3.6.6 **פיצול מפתחות וניהול בקרה כפולה של מפתחות קריפטוגרפיים**

הדרישה לפיצול מפתחות ו/או לבקרה כפולה לניהול מפתחות תמנע ממשתמש בודד לבצע לבדו משימות ניהול מפתחות כגון החלפת מפתחות או מחיקה של מפתחות. התהליכים צריכים לחייב משני משתמשים נפרדים להזין קוד – דבר אשר יוצר משני ערכים נפרדים את השרשור המהווה את המפתח הסופי.

3.6.7 **מניעת החלפה בלתי מורשית של מפתחות קריפטוגרפיים**

התהליכים אשר הוסדרו כמענה לדרישה 3.6.6 יכולים לכלול פעולות רבות למניעת החלפה לא מורשית של נתוני מפתח. בנוסף לתהליך הבקרה הכפולה, יש ליישם בקרת גישה חזקה, ביקורת ורישום של נתוני המפתחות כדי למנוע ולרשום ניסיונות לגישה בלתי מורשית.

3.6.8 **דרישה לחתימה של נאמני מפתח על טופס המציין שהם מבינים ומקבלים את תחומי האחריות שלהם כנאמני מפתח**

כדי לבצע את הפעולות המשמעותיות לניהול מפתחות המפורטות בדרישה 3.6, יש למנות נאמני מפתח מהימנים ומיומנים המבינים כיצד לבצע את ניהול מפתחות. בנוסף, על נאמני המפתח לחתום על טופס המציין שהם מבינים את תחומי

האחריות הנלווים לתפקיד זה. הדבר מוסדר בנוהל הפנימי של הארגון העוסק בהחלפת מפתחות ההצפנה של שירות התשלומים.

4. HSM

4.1 מבוא

4.1.1 HSM (Hardware Security Module) הינו רכיב קריפטוגרפי המספק יכולות הצפנה מתקדמות אשר פותח במיוחד כדי להגן על מפתחות הצפנה ועל מחזור חיי התשתית הקריפטוגרפית בארגון (ייצור, הפצה, החלפה, אחסון, השמדה וארכוב). הוא מאפשר אחסון, ניהול ועיבוד מפתחות הצפנה בסביבה מאובטחת, בתוך מכשיר מוקשח ועמיד בפני חבלה. כמו כן משמש כבסיס לבניית תשתית PKI ארגונית, הנדרשת מכל ארגון בעידן דיגיטלי פרוץ.

4.2 הנחיות

4.2.1 הייצור והאחסון של כל המפתחות אשר ישמשו להגנה על תשתיות ורכיבים רגישים ייעשה ב-HSM (זאת לעומת מפתחות לשימושים לא-רגישים אשר יכולים להיות ממקורות פסאודו-אקראיים).

4.2.2 יש העדיף ייצור ואחסון כל התעודות הדיגיטליות המותקנות בכל רכיבי אבטחת המידע ב- HSM מאובטח מרכזי – ועדיף ברכיב HSM פיזי.

4.2.3 השימוש ב-HSM מייתר את ייצור ה-PFX ואז פוחת הסיכון לאיתור המפתח כשהוא מאוחסן במקומות שונים ופחות מאובטחים.

5. תשתית ענן

5.1 מבוא

5.1.1 עולם הענן הולך ותופס נתח גדל והולך מתשתיות המחשוב ופרק זה מתייחס להצפנת תשתיות הענן תוך הפרדה בין תשתיות ענן החשופות לעולם ובין תשתיות ענן לשימושים פנימיים.

5.1.2 פרק זה מתווה קווי מדיניות כלליים ביותר להצפנת מידע בענן – בתנועה ובמנוחה – ונהלי עבודה מפורטים יותר יגובשו בהמשך.

5.2 קווי מדיניות הצפנה בתשתית ענן

5.2.1 במקרה של צורך במפתח להתקנה בשירותי ותשתיות ענן החשופות לעולם עדיף להשתמש בחתימה מסחרית על ידי גורם מאשר CA חיצוני, אמין ומוכר - לא בחתימה אשר נוצרה על ידי ספק שירות הענן כ-Root-CA ולא בחתימה אשר נוצרה על ידי מערך הדיגיטל הלאומי (למעט במקרה ונדרשת תעודת תמו"ז).

5.2.2 אם מדובר בתעודה של ספק שירות הענן אזי יש לוודא שהבקשה תהיה של מערך הדיגיטל הלאומי על מנת שלמערך הדיגיטל הלאומי תהיה שליטה מלאה על המפתח.

5.2.3 באופן כללי אין להשתמש ב-PFX - מאחר והוא ניתן להעברה/לייצוא.

5.2.4 לא מומלץ להשתמש בתעודות Self Sign. יש לעשות שימוש בתעודות מאומתות בלבד.

5.2.5 במידה והשתמשת בתעודות Self Sign אזי יש לייצר את המפתח בעצמך כדי שהתעודה במערכת תהיה שלך ולך תהיה השליטה עליה.

5.2.6 ככל שניתן, יש לעשות שימוש ב-HSM וירטואלי של ספק שירות הענן לצורך שמירת התעודות ולא לאחסן אותן בתשתיות המערכת.

5.2.7 כל חריגה מקווי מדיניות אלו חייבת קבלת אישור מערך ההגנה בסייבר ומתועדת במסמכי הפרויקט כנדרש.