

24 באפריל 2022

כ"ג בניסן ה'תשפ"ב

## OpJerusalem – המלצות התגוננות

### תקציר



1. קמפיין יום ירושלים האיראני, המוכר גם בכינויו OpJerusalem ו-AlqudsDay – מציין את הזדהות העם האיראני עם המאבק הפלסטיני.
2. הקמפיין מתקיים מדי שנה ביום שישי האחרון של חודש הרמדאן, ומתאפיין בהפגנות ברחבי איראן והרשות הפלסטינית, וכן בפעילות התקפית אנטי ישראלית במרחב הסייבר.
3. בשנים עברו, התמקדו התקיפות בניסיונות להשחתת אתרים, ובניסיונות לגרימת נזק משמעותי ברשתות מידע ארגוניות.
4. השנה הקמפיין צפוי להתחיל ב-29.04 ועשוי להמשך מספר ימים לפני/אחרי.
5. מסמך זה מפרט המלצות הגנה לשם הערכות לקראת קמפיין OpJerusalem והעלאת חוסן הגופים מפני איומי סייבר. חלק מההמלצות הן טכניות ומומלצות ליישום ע"י גורם מקצועי ומוסמך. בכל מקרה של חשד לאירוע סייבר ניתן לפנות ל-CERT הלאומי לדיווח במספר 119.

### רקע



1. מטרת הקמפיין: יצירת הד תקשורת וקידום אג'נדה אנטי ישראלית, לעורר בהלה בקרב הציבור הישראלי, חשיפת מידע אישי, השחתת אתרים, פגיעה באמון הציבור ועוד.
2. שיטות תקיפה נפוצות:
  - ניסיונות להתקפות מניעת שירות (DoS/DDos).
  - ניסיונות לחדירה למאגרי נתונים והדלפת מידע.
  - ניסיונות להדבקה בנוזקות כופרה (Ransomware).

- ניסיונות להשחתת אתרי אינטרנט (Defacement).
- ניסיונות לפריצה לשרתי ארגונים וחברות.

## המלצות התגוננות



### 1. נורמות התנהגות

- העלאת מודעות בקרב עובדים ולקוחות בנוגע לקמפיין.
- ריענון התכנית הארגונית להתאוששות מאירוע סייבר.
- שיתוף מידע ועדכונים שוטפים באמצעות מערכת סייברנט.

### 2. גיבוי ויכולת התאוששות

- גיבוי על בסיס קבוע של מידע הינו בעל חשיבות מרבית, את הגיבוי יש לבצע במדיה נפרדת אשר אינה מחוברת באופן שותף למחשב או לשרת המגובה.
- הגדרת המיקום בו יבוצע הגיבוי: התקן חיצוני נייד אותו יש לנתק לאחר הגיבוי ולשמור במיקום נפרד ומאובטח, שרת גיבוי או אחסון מבוסס ענן. במקרה זה חשוב לוודא כי שירות הענן מספק הצפנת נתונים ואימות דו שלבי.
- ככלל מומלץ לבצע גיבוי במספר ערוצים במקביל: שירותי ענן, DoK, ספריית גיבוי קלטות ושמירה על עותק גיבוי (קר).
- גיבוי תצורה לרכיבי תשתית ותקשורת, כגון נתבים, FW, מערכות אבטחה וכד'.

### 3. הגנה מפני תקיפת Dos/DDoS על אתר האינטרנט

- הגדרת מערכת WAF להפעלת Bot Mitigation, זיהוי חתימה ייחודית של התקיפה, חסימתה וכד'.
- בהעדר יכולת ארגונית מתאימה, מומלץ לבחון שימוש בשירותים מנהלים כגון WAF ו-Anti DDoS ע"י ספקית התקשורת.
- בחינת האפשרות לביצוע פעולות כגון:

- חסימת תעבורה ממדינות בעלות פוטנציאל סיכון גבוה (על בסיס Geo Location).
- במקרים חריגים של תקיפה מחו"ל – חסימה גורפת של פניות מחו"ל.
- חסימה ב-FW של כתובות הידועות כבעלות מוניטין בעייתי או עוין (על בסיס Ip Reputation).
- זיהוי וחסימת כתובות של שירותים כגון TOR או Anonymizer המאפשרים גלישה אנונימית.
- העברת האתר לענן המעניק שרותי WAF מתקדמים, במידת הצורך.

#### 4. הגנה מפני השחתת אתר אינטרנט

- עדכון גרסת CMS (Content Management Systems) כגון Joomla, WordPress, Drupal וכד', ובפרט גרסות של תוספים (Plugins) בהם מצויות רוב החולשות.
- הקשחת חיבור ל-CMS, אך ורק באמצעות חיבור TLS (Transport Layer Security) מאובטח והזדהות דו שלבית (2FA).
- הגבלת גישה לשרת ומערכת ניהול התוכן למספר כתובות ה-IP המינימליות הנדרשות.
- בדיקת תקינותם של שדות הקלט באתר ווידוא כי אינם מאפשרים הכנסת תווים שאינם נדרשים או תואמים את הערכים הצפויים.
- הפעלת ניטור אבטחתי ללוגים על שרת ה-Web, לאיתור פניות חריגות וזיהוי תקיפות בדיעבד. כמו כן, הפעלת ניטור לוגים במערכת ההפעלה לאיתור וזיהוי פעילות חריגה.
- הקשחת חוקי ה-FW והגבלת גישה בפרוטוקולים הנדרשים בלבד.
- במידת האפשר, ביצוע סריקת אבטחה לאיתור פעילות זדונית של גורמים חיצוניים, כגון: Waterhole.
- הכנת דף אינטרנט חלופי המשמש להחלפת האתר הנתקף בעת הצורך.

- בחינת האפשרות לבצע ניתוב תעבורה דרך מסננים מובנים של חברת האחסון או ספקית התקשורת.
- הגבלת שימוש בהרשאות Local Admin היכן שאין הכרח בהן.

#### 5. מניעת התחזות לבעל דומיין ו"חטיפת" האתר

- בדיקה מול ספקיות שירות ה-DNS, כי ממשק העבודה מולן מוגן ע"י מנגנון 2FA. כמו כן, יש לדרוש מהן שכל שינוי ב-Domain יחייב אישור פורמלי ממנהל ה-Domain בארגון. לשם כך יש לוודא כי פרטיו העדכניים שמורים במאגרי המידע של הספקית.
- על מנת למנוע התחזות מול רשם הדומיין מומלץ לבצע מהם לבצע פעולת Lock על ה-Domain, כך שרק מנהל ה-Domain יוכל לבצע שינויים. כמו כן, יש למסד מנגנון אימות דו-שלבי גם מול הרשמים.
- הטמעת טכנולוגיית DNSSEC, דרך רשם כתובות המתחם (Registrar Domain). במקרה זה יש לבקש מהרשם כי תעבורת ה-DNS עבור כתובות בסיומת IL תתבצע תוך שימוש ב-DNSSEC. טכנולוגיית ה-DNSSEC מאפשרת לשרת ה-DNS מולו עובד הארגון, לוודא כי התשובה אותה הוא מספק למשתמש אודות כתובת ה-IP של האתר או השירות אליו מבוצעת הגלישה היא הכתובת הנכונה.

#### 6. צמצום החשיפה לזליגת מידע ממאגרי נתונים

- יישום תהליכי Tokenization באופן שיצמצם את כפילויות הנתונים במספר מסדי נתונים שונים.
- הצפנת נתונים רגישים במסדי הנתונים.
- יישום בקורות ואיתור אנומליות במסדי הנתונים הקיימים.
- יישום מדיניות גישה מחמירה למסדי הנתונים הכוללת בין היתר:
  - הגבלת שימוש בחשבונות אדמיניסטרטיביים כגון Root/SA.
  - ניהול תהליך שוטף לשינוי סיסמאות לחשבונות אדמיניסטרטיביים כגון Root/SA.

○ יישום הרשאות מינימליות נדרשות עבור יישומים המשתמשים במסדי הנתונים.

○ יישום הרשאות מינימליות נדרשות עבור משתמשים אדמיניסטרטיביים המפתחים, מתחזקים ותומכים במסדי הנתונים.

#### 7. צמצום החשיפה לשיבוש מידע במאגרי נתונים

- יישום ותרגול תכנית הגיבויים אל מול תרחיש שיבוש נתונים.
- יישום תכנית DR הערוכה למתן מענה לתרחיש של שיבוש נתונים:
  - מיפוי מסדי נתונים קריטיים.
  - ניתוח משמעויות השיבוש.
  - אפיון מענה מדורג להתאוששות.
  - התייחסות ליעדי השירות.
  - התייחסות להערכות ארגונית (עובדים, מנהלות וכיו"ב).

#### 8. צמצום החשיפה לחדירה בערוץ הדוא"ל

- בחינת מדיניות קבלת סוגי קבצים בערוץ הדוא"ל, כמו גם בהורדת קבצים באינטרנט. מומלץ להגביל סוגי קבצים המורשים בכניסה לארגון למינימום הנדרש מבחינה עסקית, ולחסום את כניסת שאר סוגי הקבצים.
- פתיחת מסמכים תחת נטרול מאקרו וב-Protected view.
- שימוש בטכנולוגיות "ארגז חול" (Sandbox) In-line, מנועי AV ומערכות הלבנה עבור צרופות המגיעות בדוא"ל בהתאם לניהול סיכונים.
- מניעת ריצה של קבצי הרצה לא מוכרים או מאושרים בתחנות קצה.
- הטמעה שוטפת של עדכוני אבטחה בתחנות קצה הנגישות לרשת האינטרנט ולדוא"ל חיצוני.
- יישום פתרונות מתקדמים בתחנות קצה של משתמשים בעלי פוטנציאל סיכון גבוה למימוש תקיפה בערוץ הדוא"ל (כדוגמת קיבוע תצורה, הרצת קוד חתום בלבד, עבודה ממערכת הפעלה בתצורת Read Only בלבד וכיו"ב).

- ניטור אנומאליות בהזדהות לתיבות דוא"ל של הארגון, הכוללים בין היתר ניסיונות הזדהות כושלים, מדינות מהם מבוצעת הזדהות, מספר עמדות קצה או מכשירים מחוברים וסוגים, שעות התחברות.
- מימוש הזדהות חזקה לתיבות דוא"ל ומתן גישה רק באמצעות עמדות קצה / Mobile שמוחלת עליהן מדיניות וכלי האבטחה והבקרה של הגוף.
- במידה והארגון מאפשר גישה מרחוק לשרת הדוא"ל, מומלץ להגבילה באמצעות שירות VPN, ולמנוע חשיפת השרת לרשת האינטרנט.
- יישום תקן ה-DMARC, כמפורט ב**קישור**.

## 9. צמצום החשיפה להתפשטות נזקה ברשת

- מניעת פעילות שוטפת עם הרשאות גבוהות על ידי מנהלנים (כדוגמת שימוש בחשבון נפרד לפעולות מנהלתיות הדורשות הרשאות גבוהות).
- יישום כלים לזיהוי ומניעת ניצול חולשות בתווך התקשורת.
- בידול וסגמנטציה במערכות המידע.

## 10. הגנה מפני תקיפות Web Shell

- ויודא כי מערכת ההפעלה המותקנת על השרתים, ותוכנת שרת ה-WEB, מעודכנות בעדכוני האבטחה האחרונים של היצרן.
- עדכון גרסת CMS (Content Management System) כגון WordPress, Joomla, Drupal וכו', ובפרט גרסאות התוספים (Plugins), בהם מצויות רוב החולשות.
- הקשחת שרת ה-WEB בהתאם להנחיות היצרן ועל פי העיקרון של Least Privilege. מומלץ לוודא שלא הופעלו שירותים (Services) שאינם נחוצים.
- שימוש בתוכנה מסוג File Integrity Monitoring לזיהוי שינויים בקבצים על השרת.
- העלאת תוכן וקבצים לשרת עשויה להוות נקודת כשל אשר התוקף עלול לנצל לטובת השגת גישות ראשונית. מומלץ להקפיד על:
  - שימוש במערכת הלבנה בעת העלאת קבצים.

- אפשרות העלאת תוכן או כתיבה על-ידי משתמשים אך ורק לספרייה ייעודית, אשר המשתמש יכול לכתוב אליה אך לא לקרוא ממנה.
- וידוא כי סוגי הקבצים אשר ניתנים להעלאה מוגבלים על פי הדרישות העסקיות בלבד. מומלץ להימנע ככל האפשר (אלא אם יש צורך עסקי מהותי) מהעלאת קבצים אשר מאפשרים הרצת קוד.
- הקשחת הזדהות ל-CMS (Content Management System), אשר תבוצע רק באמצעות חיבור TLS מאובטח (1.2 ומעלה בלבד) והזדהות באמצעות שני אמצעי זיהוי (2FA). מומלץ לבחון אם ניתן להגביל גישת הניהול לכתובות IP ידועות מראש.
- מניעת גישה חופשית של שרת ה-WEB למשאבים רשתיים. יש לנטר גישה אל השרת וממנו ולוודא כי גישת משתמשים נעשית אך ורק בפרוטוקולים ופורטים מתאימים-HTTPS.

## 11. הגנה מפני ניצול החיבור לעבודה מרחוק

- בחינת הענקת הרשאות גישה מרחוק לתיקיות מחשוב. מומלץ להתיר גישה לתיקיות חיוניות בלבד.
- הפרדה בין גישה לדוא"ל לבין גישה לשרת, תיקיות ונכסים רגישים. אם הגישה לאחרונים נחוצה מומלץ לפתוח את הגישה לפרק הזמן הנדרש בלבד באמצעות איש המחשוב הארגוני.
- הסרת הרשאות גישה של העובדים למערכות ארגוניות/ממשקים שאינם חיוניים.
- הגדרת מדיניות אכיפת הגדרת סיסמאות מורכבות וקשות לניחוש באמצעות מנגנון ניהול המשתמשים (כגון GPO במיקרוסופט), ואילוץ המשתמש להחליף סיסמה באופן עיתי, במידת האפשר גם הגדרת OTP- one time password כאמצעי זיהוי נוסף.
- הגדרת חוקים בחוקת ה-FireWall (הארגוני והמקומי) אשר מאפשרים גישה מרחוק, כך שגישה זו תצומצם למינימום וכן כי מתקבלים לוגים לתיעוד ההתחברות. בנוסף, מומלץ להגדיר מדינות ואזורים אשר מורשים להתחבר לארגון.
- במחשב נייד/נייד- הגבלת הגישה לשורת פקודה (PowerShell) כך שלא יהיה ניתן להריץ סקריפטים שמקורם לא ידוע, או שמקורם ממחשב אחר.

- התחברות של עובדים דרך ממשק מאובטח כגון שירות VPN מרכזי עם הצפנה והזדהות חזקה מתאימה.
  - הקלטת ה-session ושמירת ההקלטה לפרק זמן קבוע (חודשים/שבועות).
  - למידע אודות גישת משתמשים לסביבה הארגונית ע"י שימוש ברשת וירטואלית פרטית (VPN) ראו [המלצות](#) של מערך הסייבר בנושא.
  - לפירוט נוסף אודות המלצות הגנה לארגונים ועסקים לעבודה מהבית שפורסמו על ידי מערך הסייבר הלאומי, [היכנסו](#).
12. המלצות נוספות ניתן למצוא ב[תורת ההגנה בסייבר לארגון](#).

## פגיעויות שפורסמו בשנים האחרונות – קיים חשש לניצולן במסגרת הקמפיין



לכלל הפגיעויות פורסם POC וישנן עדויות על ניצולן בפועל.

### 1. פגיעות הרצת קוד מרחוק ב-Spring Framework

פרויקט Spring אחראי ל-Framework הפופולרי ביותר עבור יישומי Java. בחודש מרץ 2022 פרסם פרויקט Spring מידע לגבי פגיעות העלולה לאפשר לתוקף הרצת קוד מרחוק (RCE) ראו [פרסום מערך הסייבר הלאומי](#) בנושא.

### 2. פגיעות בספריית Log4j:

ספרייה חינומית בשם Log4j, אשר מקורה בפרויקט Apache, כלולה במספר רב של מוצרים מתוצרת יצרנים שונים, ובשירותי ענן שונים. הספרייה משמשת לרישום לוגים בתוכנות שונות הכתובות בשפת Java. בחודש דצמבר 2021 פורסמה פגיעות קריטית בספרייה, המאפשרת לתוקף מרחוק הרצת קוד על שרת המפעיל ספרייה זו (RCE). ראו [פרסום מערך הסייבר הלאומי](#) בנושא.



### 3. פגיעויות קריטיות בצידוד VPN של מספר יצרנים:

- ראו [הרחבה](#) בנושא הפגיעויות בצידוד SSLVPN של חברת Pulse, אשר פורסמה על ידי ה-CERT הלאומי.
- לאחרונה פרסמה חברת Pulse התרעה לגבי פגיעות במוצר ה-VPN מתוצרתה, אשר מנוצלת בפועל על ידי תוקפים בעולם. לפגיעות זו עדיין אין עדכון אבטחה ומומלץ להשתמש במעקפים המוצעים על ידי החברה. ראו [פרסום מערך הסייבר הלאומי](#).

### 4. פגיעות בשרתי Exchange של מיקרוסופט:

- בחודש יולי 2021 פרסמה מיקרוסופט עדכון אבטחה לשרתי Exchange, ראו [התרעה](#) שפרסם ה-CERT הלאומי בנושא.

### 5. פגיעויות בשרתי Exchange:

פורסמו בחודש מרץ 2021 ומוכרות בשם ProxyLogon. הפגיעויות מאפשרות השתלטות על שרת הנגיש בפרוטוקול HTTPS מרשת האינטרנט, מנוצלות בפועל על ידי מספר רב של קבוצות תקיפה בעולם, ובמקרים רבים התוקפים מתקינים WebShells על השרת לשימור אחיזה והמשך תקיפה. מומלץ לבדוק השרת לזיהוי תקיפות, וכן להתקין בהקדם האפשרי את העדכונים. ראו [התרעת מערך הסייבר הלאומי](#).

בחודש אפריל פרסמה החברה 4 עדכוני אבטחה לפגיעויות נוספות העלולות לאפשר לתוקפים הרצת קוד מרחוק על השרת. מומלץ לבחון עדכונים אלו ולהתקינם בהקדם האפשרי. [ראו פרסום מערך הסייבר הלאומי](#).

## המלצות להתמודדות עם תקיפות Web Shell



Web Shell הינו סקריפט המותקן על ידי תוקף על גבי נכס סייבר המחצין שירותי WEB (לדוגמה: אתר אינטרנט, API או ממשק PowerShell Remoting המאפשר הרצת פקודות PowerShell ע"ג מחשב מרוחק. נכס הסייבר יכול להיות נגיש ברשת האינטרנט (המקרה הנפוץ) או נכס סייבר ברשת הארגונית.

הסקריפט מאפשר לתוקף גישה אל הנכס וביצוע פעולות שונות עליו, בהרשאות של נכס הסייבר או היישומים המותקנים עליו, כמו גם דרך יעילה להרחבת הנגישות אל תוך הרשת הארגונית, וצינור יעיל להוצאת מידע ממנה החוצה כאשר הנכס נגיש לרשת האינטרנט.

## המלצות הגנה

### 1. פעולות למניעת התקנת Web Shell

על מנת להעלות Web Shell לאתר, התוקף צריך לנצל פגיעות קיימת. לרוב פגיעויות אלו יהיו קשורות להגדרות לא מאובטחות של השרת או לחולשות אפליקטיביות. להלן מגוון המלצות למניעת תקיפות Web Shell בשרתי ה-WEB בארגון:

#### 1.1.1. עדכוני אבטחה

1.1.1.1 יש לוודא כי מערכת ההפעלה המותקנת על השרתים מעודכנת בעדכוני האבטחה האחרונים של היצרן. מומלץ שלא להשתמש בגרסאות ישנות של מערכת הפעלה אשר אינן נתמכות ואינן מכילות את עדכוני האבטחה.

1.1.1.2 יש לעדכן גרסת CMS (Content Management Systems) כגון WordPress, Joomla ו-Drupal וכו', ובפרט גרסאות התוספים (Plugins), בהם מצויות רוב החולשות, ולבצע עדכון לגרסת האפליקציה האחרונה שהופצה ועדכוני אבטחת מידע.

1.1.1.3 זמן התקנת עדכונים מומלץ מרגע הפרסום הינו 24 שעות לכל היותר.

#### 1.2. הקשחה

1.2.1 יש להגדיר את שרת ה-WEB ולהקשיחו בהתאם להנחיות היצרן ועל פי עיקרון של Least Privileged.

1.2.2 יש להגדיר את שרת ה-WEB ולהקשיחו בהתאם להנחיות היצרן ועל פי עיקרון של Least Functionality, מומלץ לוודא שלא הופעלו שירותים (Services) שאינם נחוצים, וכן לא קיימים רכיבי תוכנה שאינם נדרשים לעבודה השוטפת.

להרחבה בנושא זה מומלץ לעיין במסמך "המלצות ליישום - הקשחת מערכות מחשוב".

1.2.3. יש לבצע בדיקת הרשאות עיתית למערכת הקבצים של נכס הסייבר וזאת במטרה לגלות ולזהות הרשאות חריגות, המאפשרות גישה שאינה מורשית הן בגישה מקומית והן בגישה מרחוק.

1.2.4. יש לעשות שימוש באמצעים טכנולוגיים להגבלת הגישה, כך שגם אם תוקף הצליח לגשת לספריות אתר ה-WEB, הוא עדיין לא יוכל לגשת לאזורים נוספים במערכת ההפעלה. בסביבת לינוקס לדוגמה, ניתן לעבוד עם SELinux Enforcing Mode או AppLocker.

### 1.3. אבטחת תהליך העלאה/הורדה של קבצים

1.3.1. העלאת תוכן וקבצים לשרת עשויה להוות נקודת כשל אשר התוקף מנצל לטובת השגת נגישות ראשונית.

1.3.1.1. מומלץ להשתמש במערכת הלבנה בעת העלאת קבצים.

1.3.1.2. במידה ושרת ה-WEB מאפשר העלאת תוכן או כתיבתו על ידי משתמשים, מומלץ לאפשר העלאה לאחר בדיקה אך ורק לספריה ייעודית ובמחיצה נפרדת מזו שהאפליקציה ומערכת ההפעלה מותקנות בה.

1.3.1.3. יש לוודא כי השרת מעניק שם ייחודי (GUID) לקובץ, וזאת ללא הסתמכות על מידע אשר סופק ע"י המשתמש.

1.3.1.4. יש לוודא כי הקובץ עונה לגודל וכן לפורמט (True Type) בהתאם לרשימת מותרים (Allow List). הערה: יש להימנע ממתן אפשרות לטעינת קבצי Executable.

1.3.1.5. יש לוודא כי סוגי הקבצים אשר ניתנים להעלאה מוגבלים לדרישות עסקיות בלבד. ככל שלא נדרשים קבצים אשר מאפשרים הרצת קוד מומלץ לחסום אפשרות להעלאת קבצים מסוג זה.

1.3.2. בעת תהליך הורדת קבצים / העברת קובץ ללקוח יש לבצע בדיקת הרשאות מקדימה בצד השרת וזאת במטרה למנוע נגישות שאינה מורשית לקבצים.

1.3.3. להרחבה נוספת מומלץ לעיין בקישור הבא.

1.4. דגשים בעבודה עם CMS

1.4.1. מומלץ להקשיח חיבור ל-CMS (Content Management System) - יעשה רק באמצעות חיבור TLS מאובטח (1.2 ומעלה בלבד) והזדהות באמצעות שני אמצעי זיהוי (2FA).

1.4.2. יש להגביל גישה לשרת ולמערכת ניהול התוכן למספר כתובות ה-IP המינימליות הנדרשות. ככלל, יש להעדיף מתן גישה מכתובות IP סטטיות בלבד.

1.4.3. יש להגביל את הספים והרפים בפעילות משתמש מול האתר.

1.4.4. אין לאפשר לשרת ה-WEB גישה חופשית למשאבים רשתיים, יש לנטר גישה אל השרת וממנו ו לוודא כי גישת משתמשים נעשית אך ורק בפרוטוקולים ופורטים מתאימים - HTTP (80) או HTTPS (443).

1.5. סגמנטציה ובקרת גישה

1.5.1. יש לוודא כי תעבורה יוצאת מן השרת תאפשר אך ורק אל שרתים שיש לו צורך בגישה אליהם כגון שרת DB ובפורט מתאים בלבד.

1.5.2. שרת ה-DB שבשימוש שרת ה-WEB לא ימוקם בתוך הרשת הארגונית אלא ב-DMZ וגם אליו וממנו תהיה בקרת תעבורה קפדנית.

1.5.3. יש לוודא כי שרת האפליקציה ושרת ה-DB מותקנים ברשתות ייעודיות ונפרדות (לדוגמה VLAN100 ו-VLAN200).

1.5.4. יש לוודא כי התעבורה בין שרת האפליקציה מ/אל שרת ה-DB עוברת דרך FW.

1.5.5. יש למנוע מצב שבו שרת האפליקציה / DB יכול לבצע ייזום פניה החוצה לאינטרנט (במקרים חריגים יש לאשר פרטנית, ובכל מקרה יש להימנע ממתן גישה לשרתים אלו לאינטרנט ולפיכך במידה וקיים צורך מסוג זה עדיף לעשות שימוש בשרת ממשקים ייעודי).

1.5.6. יש לחסום את האפשרות לביצוע Directory Browsing.

1.6. שימוש ב-Web Application Firewall – WAF

- 1.6.1 WAF הינו ציוד או שירות ייעודי לניטור ובקרה על תעבורה לשרתי WEB. מומלץ לעשות שימוש ב-WAF על מנת לסכל ככל האפשר את התקיפה הראשונית המאפשרת התקנת Web Shell וכן פניה של התוקף להפעלתו.
- 1.6.2 יש לוודא אכיפה ב-WAF לפי רשימה לבנה (Whitelist) של פרמטרים (לדוגמה סוג מידע-מחוזות, מספר), מס' תווים (לדוגמה עד 8 תווים), סוג תווים (לדוגמה אותיות גדולות/קטנות, מספרים, תווים מיוחדים וזאת ביחס למיקום הופעתם בקלט הרצוי), ולא להסתמך על חתימות בלבד. בהתאם לפרסומים מודיעיניים ניתן ללמוד כי תוקפים בזירה מסוגלים לעקוף חתימות של WAF, לפיכך נדרש ליישם אכיפה ע"פ רשימה לבנה.
- 1.6.3 כל שינוי בפרמטרי העבודה מחייב אישור מתאים מצד ממונה הגנת המידע בארגון.
- 1.7 פיתוח מאובטח
- 1.7.1 שמירה על עקרונות פיתוח מאובטח ומניעת פגיעויות קוד אשר מאפשרות ניצול על ידי תוקף להעלאת קבצים בלתי מורשית. מומלץ לבצע סריקת קוד לאיתור פגיעויות, במיוחד מהסוגים הבאים:
- 1.7.1.1 Cross Site Scripting (XSS)
  - 1.7.1.2 SQL Injection
  - 1.7.1.3 SSRF
  - 1.7.1.4 Remote File Inclusion (RFI) and Local File Inclusion (LFI)
  - 1.7.1.5 חשיפת ממשקי ניהול לרשת האינטרנט
- 1.7.2 יש לבדוק את תקינותם של שדות הקלט באתר ולוודא כי אינם מאפשרים הכנסת תווים שאינם נדרשים או תואמים את הערכים הצפויים.
- 1.7.3 יש לוודא כי ברשות הארגון תהליך פיתוח מאובטח בהתאם למתווה המתואר במסמך "פיתוח מאובטח – עבודת מנהל הגנת סייבר CISO עם גופי הפיתוח בארגון" של מערך הסייבר הלאומי המופיע [בקישור](#).
- 1.7.4 יש לבצע בדיקות עיתיות לאתר בהתאם למסמך OWASP Testing Guide בגרסתו האחרונה הקיים [בקישור](#).

הערה: בעת ביצוע בדיקות חוסן יש לוודא כי תיחום הסקר כולל התייחסות לבדיקות הרלוונטיות מהרשימה לעיל.

1.7.5 ישנה חשיבות לשמירה על רמת עדכניות גבוהה של ספריות קוד פתוח בהן הארגון וספקיו עושים שימוש. ראוי לציין כי חלק לא קטן מן החולשות המאפשרות הטלת Web Shell זכו לעדכון ע"י הקהילה, אך ארגונים לא ביצעו את העדכון בפרק זמן הולם (ימים בודדים מפרסום).

1.7.6 מערך הסייבר הלאומי, בשיתוף קרן רש"י, פרסמו קורס ללא עלות שמטרתו לשפר את רמת ההיכרות עם תקיפות WEB, ומומלץ להיעזר בו, [הקישור לקורס](#).

## 2. פעולות לזיהוי Web Shell

לעתים, על אף פעולות המניעה שננקטו, עשויים תוקפים לנצל חולשות חדשות ולא מוכרות (Zero-Day), לכן כמעגל הגנה נוסף יש לבצע פעולות לזיהוי Web Shell. להלן המלצה למגוון פעולות לזיהוי Web Shell בשרתי ה-WEB בארגון:

### 2.1. ניטור אירועים

2.1.1 יש לנטר אירועים חריגים בשרת, כגון הפעלה של Command Line (BASH//POWERSHELL CMD) או פקודות אחרות שאינן מתוכננות לשימוש בשרת.

2.1.2 ניטור של אחוז שימוש גבוה ב-CPU כאשר התעבורה לאתר אינה גבוהה.

2.1.3 ניטור שינוי בקבצים בשרת, הוספה של קבצים חדשים או שינוי בקבצים קיימים שלא על ידי מנהלן מוסמך של הארגון.

לזיהוי השינויים הללו ניתן להיעזר גם במערכות מסוג – File Integrity Monitoring (FIM) וזאת תוך השוואה עיתית של המצב בפועל ביחס ל-Baseline מאושר. את הבדיקה יש לבצע בזמן אמת וכן בדיקה עיתית מדי שעה. במידה ומערכת ניהול התוכן של האתר תומכת בזיהוי שינויים כאלו, ניתן להיעזר בה.

2.1.4 זיהוי קבצים בעלי חתימת זמן מאוחרת יותר ממועד העדכון האחרון של השרת.

2.1.5 איתור ניסיונות גישה לא מורשים לאינטרנט.

2.1.6. איתור קיומם של Process-ים לא מוכרים לרבות Process-ים משניים. לטובת העניין יש לעשות שימוש ב-EDR/XDR תוך השוואה ל-Baseline מוגדר.

2.1.7. יש לבצע הרצה עיתית של הכלי "עמלץ", אשר פותח ע"י מערך הסייבר הלאומי, על מערכת הקבצים המאחסנת את האפליקציה של האתר, התכנים ומערכת ההפעלה.

## 2.2. ניטור ולוגים

2.2.1. יש להפעיל ניטור אבטחתי ללוגים (Logs) על הנכסים המעורבים לאיתור פניות חריגות, ובכדי לאפשר יכולת זיהוי תקיפות בדיעבד. יש להפעיל ניטור לוגים (Logs) במערכת ההפעלה לאיתור וזיהוי פעילות חריגה. לדוגמה:

2.2.1.1. פניות והזדהות משרתים ותחנות ברשת הארגונית אל שרת ה-WEB, שלא בהתאם לאפיון המקורי של האתר

2.2.1.2. שימוש בפקודות למעבר ספריות (Directory Traversal)

2.2.1.3. גישה ישירה אך ורק ל-URL מסוים

2.2.1.4. ניתוח סטטיסטי של פניות ל-URL בדרך כלל יהיו מגוון של USER AGENTS. בפניות ל-Web Shell אמור לגשת רק התוקף, לכן צפוי שימוש ב-USER AGENTS - בודדים או ייחודיים.

2.2.1.5. פניות חוזרות ל-URL מסוים ללא שדה REFERER עלול להצביע על שימוש ב-Web Shell.

2.2.1.6. כל הפעולות המתבצעות במערכת הקבצים ובמסד הנתונים.

2.2.1.7. כל הפעולות הכוללות שינוי הגדרות תצורה.

2.2.1.8. כל החיוויים אשר עשויים להעיד על פעילות זדונית בהתאם ל-MITRE ATT&CK. יש לוודא כי מערך הניטור כולל תמיכה מובנית ביכולת זו בגרסה האחרונה וזאת ללא צורך במעורבות אנושית לאיתור חיוויים. קישור לפרסום MITRE הרלוונטי.

2.2.1.9. איתור IOCs בזמן אמת ובאופן עיתי, וזאת בהתאם להתרעות המפורסמות ע"י מערך הסייבר הלאומי וגופי מודיעין מקובלים.

2.2.1.10. ביצוע סריקת פורטים ושירותים עיתית מחוץ לרשת, וזאת במטרה לזהות חולשות ופורטים פתוחים אל העולם ללא צורך עסקי.

2.2.1.11. בנוסף יש לוודא ניטור של כלל האירועים בהתאם להמלצות מסמכים מקובלים דוגמת "[Detect and Prevent Web Shell Malware](#)".

2.2.2. ככלל, יש לוודא שמירה של הלוגים מחוץ לנכס הסייבר אשר יצר אותו למשך תקופה של 90 ימים לכל הפחות. ראוי לציין כי דרישות חקיקה ורגולציה דוגמת תקנות הגנת הפרטיות (אבטחת מידע) עשויות לחייב שמירת לוגים לתקופה ארוכה יותר, לפיכך יש לפעול ע"פ ההנחיה המחמירה יותר.

2.2.3. יש לוודא קיומו של ניטור 24/7 על מצב האבטחה של נכסי הסייבר.

### 2.3. שימוש בכלים ייעודיים

2.3.1. במידה שמתקן על השרת מנוע AV, עשוי גם הוא לסייע בזיהוי של Web Shell, אם כי אחוזי הזיהוי אינם גבוהים בדרך כלל.

2.3.2. קיימים כלים ייעודיים המאפשרים חיפוש אחר Web Shell בשרת, כלים אלו עלולים לעיתים לתת זיהויים שגויים.

2.3.3. קיימים מאגרים של חוקי YARA המיועדים לזיהוי Web Shell. ניתן להריץ חיפוש YARA על ספריות השרתים באמצעות חתימות אלו.

### 3. גיבוי והתאוששות

במידה ונמצא בארגונכם Web Shell, הארגון יידרש להתקנה מחדש של השרתים והאתר. על מנת להיערך יש לבצע את הפעולות הבאות:

3.1. יש לוודא ביצוע יומיומי של גיבוי חומרים בעלי חשיבות כגון: אתר האינטרנט, בסיסי נתונים, שרתים וכיו"ב. ביצוע הגיבוי נועד לאפשר שחזור מהיר של מידע במידת הצורך.

3.2. יש לוודא שמירה של הגיבוי על מדיה נפרדת, במיקום נפרד ומאובטח. ככלל, מומלץ לבצע גיבוי במספר ערוצים במקביל (שירותי ענן, שרתי גיבוי, קלטות וכד').

3.3. יש לבצע התקנה מחדש של השרת ושחזור האתר ממדיה בדוקה על מנת לוודא שחזור ללא Web Shell.





3.4. יש לוודא שינוי כל סיסמאות הגישה כולל סיסמאות הגישה בשרתים אליהם ניגש השרת הנגוע כגון שרתי DB.

3.5. יש לוודא עדכניות נוהל התאוששות ושחזור לגרסה תקינה בעת הצורך, במידת האפשר מומלץ לבצע בדיקה ולוודא שהשחזור עובד.

3.6. ככלל יש לשמר יכולת לבצע שחזור מאפס של נכסי הסייבר והמידע וזאת תוך פרק זמן קצר ככל הניתן.

3.7. להרחבה מומלץ לעיין במסמך של מערך הסייבר הלאומי "[שילוב עקרונות הגנת הסייבר בתהליכי גיבוי ושחזור](#)".