**Microsoft Security Bulletin Analysis**

**דוח פריסה מוצג על ידי בר כהן וסהר חיים יעקב.  
  
דוח זה עוסק בשלב הפריסה של תוצאות ניתוח הנתונים בפרויקט. מטרתו היא לתאר את תהליך יישום המודלים הנבחרים במערכת, תוך התייחסות לשילובם בפלטפורמות קיימות, ניטורם השוטף ותחזוקתם לאורך זמן, לשם שמירה על ביצועים מיטביים ודיוק תחזיות בסביבה הארגונית.**

***שם מרצה : מר אבי זכאי.***

***שם מנחה : מר חנן לב.***

***מגישים :***

***בר כהן 208110254***

***סהר יעקב 314741851***



תוכן עניינים

[1. תוכנית פריסה - 3 -](#_Toc204423977)

[GitHub 1.1 - 3 -](#_Toc204423978)

[Tableau Public 2.1 - 4 -](#_Toc204423979)

[MySQL 3.1 - 5 -](#_Toc204423980)

[Gradio 4.1 - 6 -](#_Toc204423981)

[2. תכנון מעקב ותחזוקה - 7 -](#_Toc204423982)

[1.2 אילו גורמים / השפעות צריכים להיות במעקב ? - 7 -](#_Toc204423983)

[2.2 כיצד נמדוד וננטר את תוצאות ודיוק המודלים ? - 7 -](#_Toc204423984)

[3.2 כיצד נקבע מתי פג התוקף של כל דגם ? - 7 -](#_Toc204423985)



# **1. תוכנית פריסה**

שלב הפריסה (Deployment) מהווה נקודת מפנה קריטית בכל פרויקט ניתוח נתונים ולמידת מכונה. לאחר תהליכי איסוף הנתונים, עיבודם, בחירת המודלים ואימונם, מגיע השלב שבו התובנות והפתרונות המפותחים הופכים לנגישים ושימושיים עבור קהל היעד. פרק זה יתאר את תוכנית הפריסה המקיפה של תוצרי הפרויקט, תוך התייחסות למספר ערוצי יישום ושיתוף, אשר יבטיחו את שימור העבודה, הצגתה באופן מקצועי ויצירת ממשקים אינטראקטיביים עם המודלים שפותחו.

*בפרק זה, נשלים את תוכנית הפריסה באמצעות ביצוע הפעולות הבאות:*

**שימור וניהול קוד באמצעות: GitHub**

נבצע העלאה מסודרת של כל מסמכי הפרויקט, קוד המקור, וקבצים נלווים למאגר GitHub. פעולה זו תבטיח שימור גרסאות יעיל, שקיפות תהליכית, ותאפשר הצגה מפורטת של העבודה בפני גורמים רלוונטיים, תוך קידום שיתוף ידע ופוטנציאל לשיתופי פעולה עתידיים.

**ניהול נתונים וחקר בMySQL - :**

הנתונים המעובדים והמנוקים יועלו למסד נתונים .MySQL מסד הנתונים ישמש כמאגר נתונים מרכזי ויציב, ויספק פלטפורמה עוצמתית לניהול הנתונים ויאפשר ניתוח וחקר מעמיק באמצעות כלי BI שונים וניתוח נוספים שאינם Tableau Public ,לדוגמא Power BI. יכולת זו חיונית לגמישות אנליטית ולתמיכה בנפחי נתונים עתידיים.

**הצגת תובנות חזותיות באמצעות : Tableau Public**

לצורך הצגת התובנות החזותיות העיקריות מניתוח הנתונים, נציג לוח מחוונים (Dashboard) איכותי ואינטראקטיבי באמצעות פלטפורמתTableau Public . הנתונים המשמשים לדשבורד זה יועלו ישירות ל Tableau Public וזאת לאחר שהנתונים עברו את שלבי הניקוי והעיבוד הנדרשים. לוח המחוונים ימחיש את הממצאים המרכזיים ויאפשר למשתמשי קצה לחקור את המידע בצורה ויזואלית ואינטואיטיבית.

**הנגשת המודל הסופי באמצעות : Gradio**

נציג את תרחישי השימוש של המודל הסופי שבנינו, אשר מתבסס על אלגוריתם .CatBoost לשם הדגמה והנגשה, נפרוס את המודל באמצעות דף אינטרנט ייעודי שנבנה באופן מודולרי, תוך שימוש בספריית Gradio שבפייתון. ממשק זה ישלב עיצוב ויזואלי מתאים (באמצעות (CSS ויאפשר למשתמשים אינטראקציה ישירה עם המודל, קבלת תחזיות והבנת אופן פעולתו בפועל.

פרק זה ידגיש כיצד הטמעת הפתרונות הללו אינה רק סיום של תהליך הפיתוח, אלא פתיח לשלב חדש שבו הערך העסקי והתפעולי של הפרויקט ממומש במלואו.

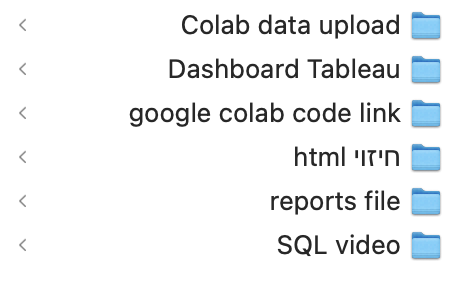
## ‪With GitHub - PyXAI documentation‬‏GitHub 1.1

כדי שהמודל ונתוני הדוחות ישמרו לאורך זמן ויהיו זמינים ותקפים אנו נצטרף לפלטפורמת ה GitHub על מנת שנוכל להעלות בה את מסמכי הפרויקט. GitHub היא הפלטפורמה האידאלית לכך, היא מאפשרת שמירה וגיבוי עם שיתוף פעולה בין צוותים, **הצגת הפרויקט לקהל מקצועי או אקדמי**, ותיעוד תהליכי העבודה באופן מסודר ונגיש. בנוסף דרך GitHub ניתן למשוך את המודל לצורך ניסוי וסקירה וגם הטמעה לאותן החברות שיצטרכו מודלים אלו.

* **שמירה וגיבוי בענן** – כל שינוי נשמר ומגובה אוטומטית, עם אפשרות לשחזור גרסאות קודמות.
* **שיתוף פעולה בין חברי הצוות** – כל חבר צוות יכול לגשת לתוכן, להציע שינויים, לפתור קונפליקטים ולעבוד בסנכרון מלא.
* **תיעוד מקצועי** – ניתן לתעד את תהליכי העבודה בצורה מסודרת.
* **חשיפה לקהל מקצועי** – אנשי מקצוע, חוקרים, או מראיינים פוטנציאליים יכולים להתרשם מהפרויקט ומהיכולות הטכניות והמתודולוגיות של הצוות.
* **סטנדרט בתעשייה** – שימוש ב־Git ובפלטפורמות כמו GitHub הוא חלק בלתי נפרד מתהליכי הפיתוח בתעשיית ההייטק. שילוב של Git בפרויקטים אקדמיים מדמה סביבת עבודה אמיתית, ומכין את חברי הצוות לעבודה בצוותי פיתוח מקצועיים, תוך שימוש ב־Branches, Pull Requests, Code Reviews וניהול גרסאות.

כדי להעלות את המסמכים בצורה מסודרת אנו ניצור תיקיות כדי לאחסן כל קובץ בנפרד למשל -

בתוך כל תיקיה נאחסן את המסמכים הרלוונטיים.



בנוסף לשמירה גיבוי והצגה של מסמכי הפרויקט , GitHub תורם להרחבה עתידית, שקיפות תהליך כמו מעקב ברור ושיתוף פעולה על סמך קוד פתוח.

לסיכום, GitHub הוא הרבה מעבר למקום אחסון: הוא תשתית שלמה לניהול ידע, תהליכים, שיתוף פעולה ומקצועיות, והוא מעניק לפרויקט יתרון אמיתי – גם בטווח הקצר של ההגשה, וגם בטווח הארוך של יישום עתידי בעולם האקדמי או העסקי.

## ‪Tableau Expert Consulting Services by Select Distinct‬‏Tableau Public 2.1

כדי להמחיש את הנתונים שהורדנו מאתר מיקרוסופט , הכנו בTableau - ויזואליזציה Dashboard מותאמת לנתונים.  
כדי להמחיש בצורה ויזואלית את הנתונים והמשמעות שלהם יצרנו גרפים שונים , עמודות , קווים , טבלאות ועוד.

בתמונה מתוך הDashboard - שהצגנו ניתן לראות צילום מסך חלקי ובו חילוק הנתונים לפי ימים ,וסוגי תקיפה.

בצילום המסך סיננו את התקיפות ליום שלישי וסוג התקיפה מסוג - Important.

ניתן לראות את הכלי בה בוצע התקיפה - מניעת שירות, העלאת הרשאות, חשיפת מידע, הרצת קוד מרחוק, עקיפת אבטחה, עקיפת תכונות אבטחה, זיוף, ועריכה בלתי מורשית.

בנוסף ניתן לראות את כמות התקיפות שבוצעו ביום הזה , ממוצע התקיפות , מגמת התקיפות ועוד.

ניתן לומר שביום שלישי בוצעו רוב התקיפות כ -9757, מסוג Important.

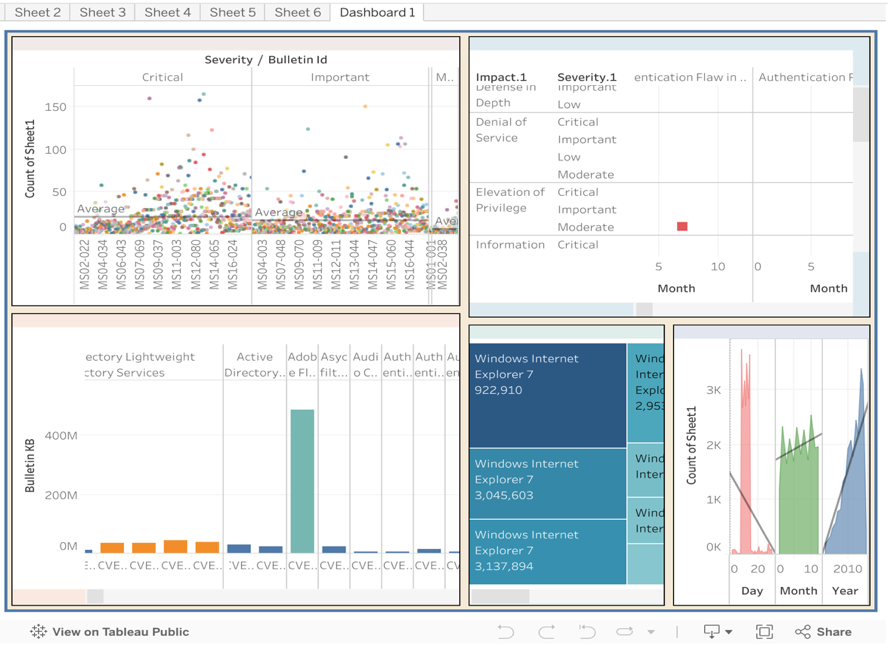
ניתן להסיק כי יום שלישי הוא היום בו מיקרוסופט נמצאת בסכנה מפני רוב התקיפות ולכן כדאי לנצל את רוב המשאבים על יום שלישי.

בגרף הגודל של העיגולים, הגודל מציין את כמות התקיפות וגם ניתן לראות מאיזה סוג מערכת הפעלה בוצעה התקיפה, כלומר - mac/windows.

צילום המסך הוא מתוך תהליך הכנת ה - Dashboard ולא התוצר המוגמר.



[Dashboard Link](https://public.tableau.com/app/profile/bar.cohen4006/viz/Book1_17535246082780/Dashboard1)



## ‪Python with MySQL CRUD Operations: A Practical Guide » Khmer168.com - Notes for sharing IT Knowledge, Comparison, Best Practices, and ICT Tips in Cambodia RatanaSOTH - the IT Geek in Cambodia‬‏MySQL 3.1

בחרנו לעלות את הנתונים שלנו לתוך מסד נתונים MySQL , באמצעות קוד פיתון.

כדי להתממשק צריך פרטי הזדהות, הקמנו מסד מתוך הפיתון בשם project\_database ושם הטבלה שבחרנו Microsoft\_Attack , בנוסף הוספנו מפתח ID מתוך הפיתון אם לא קיים והצלחנו להריץ שאילתות פשוטות יחסית.

הקוד פועל על התוכנה באופן מקומי ישירות על מחשב האישי, ללא צורך בהתקשרות עם שרתי ענן חיצוניים. גישה זו מאפשרת שמירה מירבית על פרטיות הנתונים, מהירות תגובה גבוהה יותר, ושליטה מלאה על סביבת העבודה. כמו כן, עבודה מקומית מבטיחה עצמאות מרשתות תקשורת ומפחיתה סיכונים.

הנתונים שהורדנו אינם מכילים את עמודת ה - ID, היא נוספה לשם מפתח ייחודי של כל תקיפה.

באמצעות ה – SQL, הרצנו שאילתות מתקדמות יחסית , כלי העזר הזה איפשר לנו לסנן ולמצוא פרטים חשובים על הנתונים (סינון מתקדם יותר).

לפי צילום המסך אנחנו בוחרים בעמודות id , הרכיב שהושפע , שיחזור , סיווג ראשוני ומשני של התקיפה. בוחרים מתוך הטבלה בשם חלופי ma (**M**icrosoft\_**A**ttack), כדי לפשט את התהליך. מתוך שורות אלה סיננו את הרכיבים של windows ו- mac שמכילים את המספר 16, לבסוף סידרנו את הנתונים לפי סדר אלפביתי יורד מהסוף להתחלה לפי משתנה המטרה - Severity (Z-A) .

ערכנו השוואה בין windows וmac - – מחשבי Mac לא מצריכים הפעלה מחדש לאחר עדכון לעומת Windows שכן צריכים הפעלה חדש, בנוסף ניתן לראות כי מרבית התקיפות של Windows קריטיות בסיווג הראשוני ובמשני לעומת Mac שנחשבות חלשות, אפשר להסיק מכך על רמת האבטחה ותחזוקת המערכות השונות.

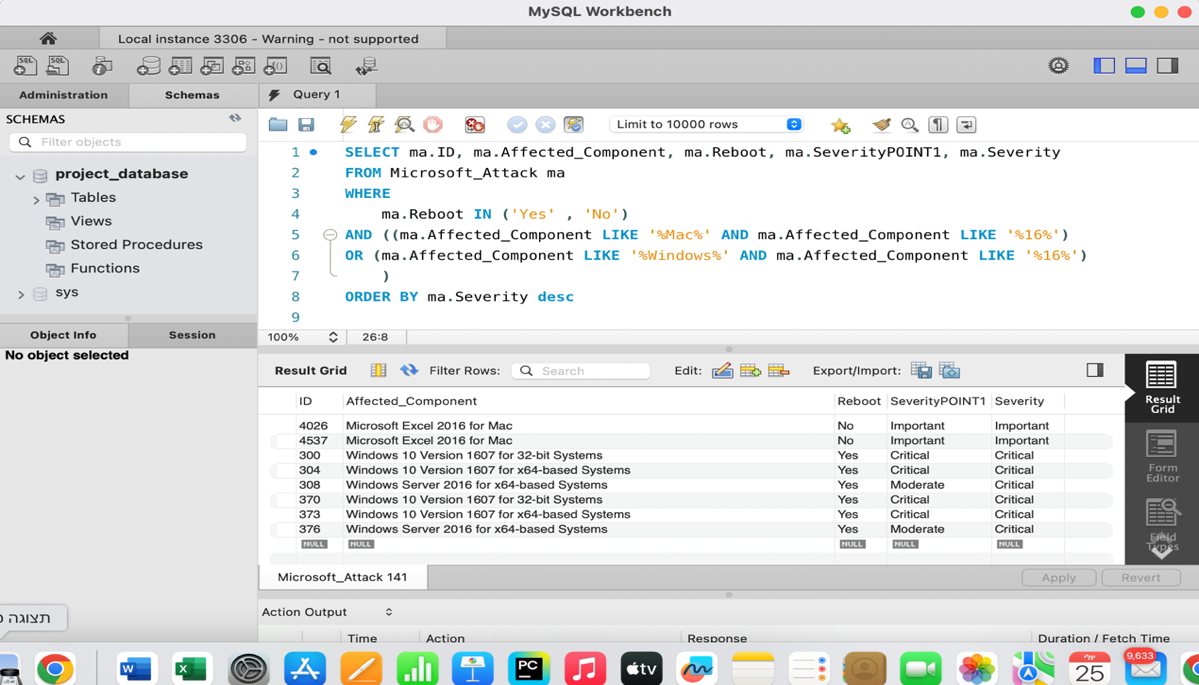
באופן כללי, השימוש ב-SQL מאפשר לנו לנהל את הנתונים בצורה חכמה ומדויקת, להוציא מהם תובנות איכותיות ולייעל את תהליך העבודה עם מאגרי מידע מורכבים.

למה SQL?

בSQL ישנם שאילתות שמאפשרות מיקוד וסינון נתונים בצורה מדויקת ומהירה, מה שחוסך זמן רב בעבודה עם כמויות גדולות של מידע (כמו בפרויקט זה). במקום לעבור על כל הרשומות באמצעות כלים פחות גמישים ולא יעילים, ניתן לכתוב שאילתה שתחלץ בדיוק את המידע הרלוונטי בלבד.

בנוסף, SQL תומכת בפעולות חישוביות וכלליות שמאפשרות לבצע סיכומים, ממוצעים, ספירות ועוד, כך ניתן לקבל דוחות וניתוחים סטטיסטיים בצורה קלה ומהירה. פעולות אלו מאפשרות לזהות מגמות, חריגות ודפוסים בנתונים במהירות, ומשפרות את איכות הניתוח העסקי והמחקרי.

SQL נחשבת לשפה עוצמתית מאוד עם יכולות מורכבות לביצוע חיפושים, סינונים, מיזוג טבלאות, חישובים ודוחות, בעוד שמערכות ניהול נתונים אחרות, כמו NoSQL, מתמקדות בגמישות מבנית ולא תומכות בשאילתות מורכבות ברמת היעילות.

בנוסף כלי הSQL הוא בחירה מותאמות עבור מחקר של למידת מכונה ולצוות דאטה אנליסט.

## תמונה שמכילה עיצוב תוכן בינה מלאכותית גנרטיבית עשוי להיות שגוי.‪gradio · PyPI‬‏Gradio 4.1

כדי להמחיש את תוצאת המודל CatBoost בניתוח הנתונים, פיתחנו כלי אינטראקטיבי בפיתון באמצעות Gradio (ייצור דף אינטרנטי) , המאפשר חיזוי רמת הסיכון של רמת האבטחה במוצרי מיקרוסופט.

באמצעות פלטפורמת Gradio ניתן לבנות ממשק אינטרנט HTML מתוך Python באמצעות ספריית **Dash**, ספריית **Dash** מכילה רכיבים שיכולים לבנות אתר אינטרנט כמו Html וגם עיצוב חיצוני/פנימי מתאים.

ספריית Dash

**Dash** היא ספריה שניתן ליצור באמצעותה יישומים אטרקטיביים בצורה קלה**,** שימושה העיקרי הוא בתחום **ניתוח הנתונים, למידת מכונה, מדעי נתונים** וויזואליזציה.  
יתרון הבולט של הספרייה הוא שאין צורך בידע קודם בHTML, CSS או JavaScript.

*ישנם מספר יתרונות בולטים לשימוש ב Dash :*

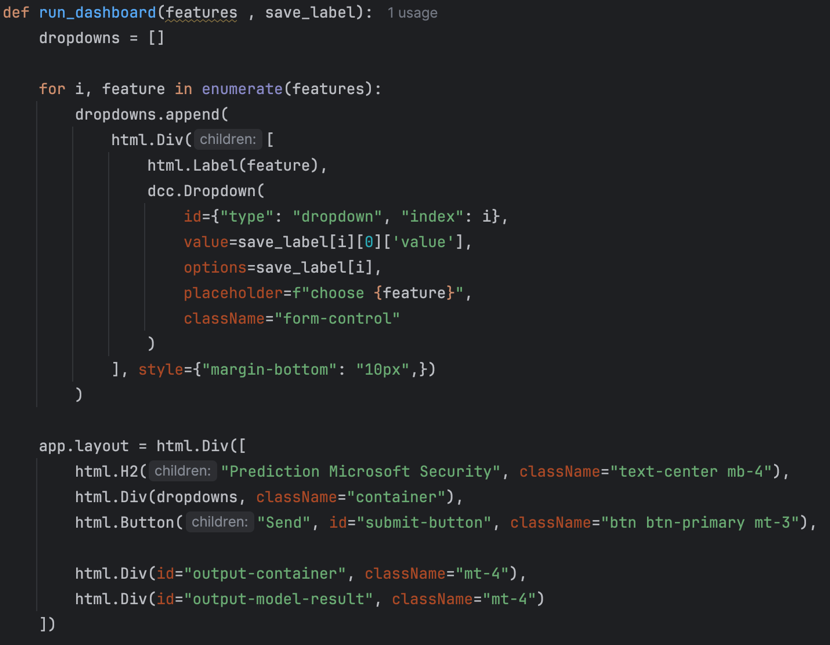
* **קוד נקי** , באמצעות python בלבד.
* **מותאם למידול נתונים וגם ללמידת מכונה.**
* **הרצה מקומית או על שרת ענן – לכן יש צורך להריץ את הקוד מחדש בכל פעם.**

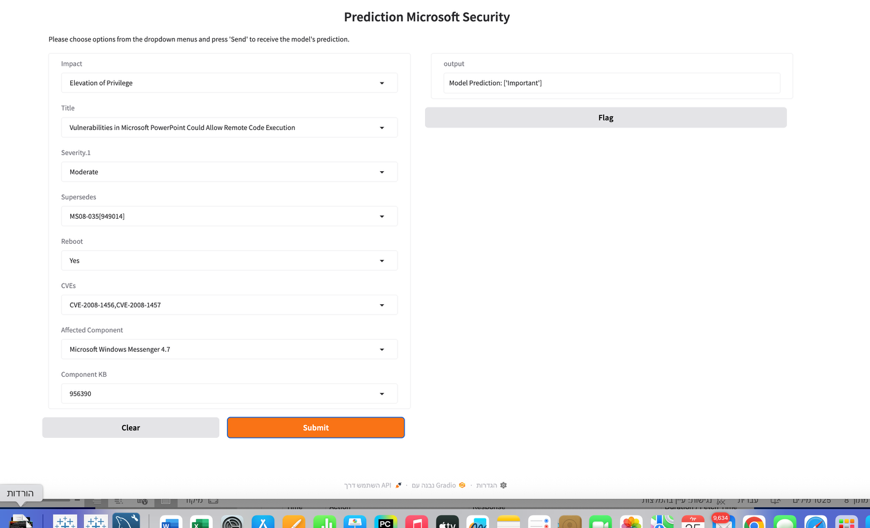
|  |  |  |
| --- | --- | --- |
| תכונה | Dash | Gradio |
| מטרה | ויזואליזציה וDashboards | מודלי למידת מכונה וai. |
| שפות | Python\html\css | python |
| עיצוב | Css + boostrap | בסיסי |
| התאמה | אפליקציות מורכבות | הדגמות מהירות ונוחות |

בחרנו בDash וGradio כי ניתן לכתוב בשפת פייתון ולא לערבב גרסאות שונות של הפרויקט .

בדוגמא זו הקוד הוא מודולרי ולא עבור תכונות ספציפיות , הפרמטרים הם רשימת התכונות ורשימת האפשרויות הייחודיים מתוך כל עמודה.  
  
*ניתן לבחור את התכונות והפרמטרים המתאימים ע״י לחיצה ולקבל משוב , התכונות (תכונות המודל) הם :*

1. **Impact:** Elevation of Privilege  
   השפעת הפגיעות שנבחרה היא העלאת הרשאות. המשמעות היא שהתוקף יכול, באמצעות ניצול הפגיעות, להשיג רמות גישה גבוהות יותר במערכת ממה שהיו לו במקור, לדוגמה ממשתמש רגיל למנהל מערכת.
2. **Title:** Vulnerabilities in Microsoft PowerPoint Could Allow Remote Code   
    כותרת הפגיעות מתארת בעיות אבטחה.
3. **Severity.1:** Moderate  
    חומרת הפגיעות ראשונית סווגה כ"בינונית".
4. **Supersedes:** MS08-035 (940214)   
   עדכון אבטחה שמחליף או מבטל עדכונים קודמים תחת הקוד.
5. **Reboot:** Yes  
   האם יידרש אתחול (הפעלה מחדש) של המערכת לאחר יישום התיקון לפגיעות זו, על מנת שהשינויים ייכנסו לתוקף באופן מלא.
6. **CVEs:** CVE-2008-1456, CVE-2008-1457  
    מזהה ייחודיים.
7. **Affected Component:** Microsoft Windows Messenger 4.7  
    הרכיב המושפע.
8. **Component KB:** 956380  
    מידע נוסף ופרטים טכניים.

לאחר בחירת הפרמטרים מתוך הרשימות שהגדרנו, נריץ את המודל ותוצאת המודל היא, שהתקיפה חשובה.  
כלי זה ממחיש כיצד ניתן להשתמש בפיתון וב - Gradioלבניית יישומים אינטראקטיביים המאפשרים למשתמשים לחקור ולנתח נתונים בצורה נוחה ודינמית.

תצוגה מתוך הדף שיצרנו.

# **2. תכנון מעקב ותחזוקה**

בפרק זה, נתמקד בשלבי הסיום הקריטיים של הפרויקט, הכוללים תוכנית מפורטת למעקב שוטף, ניטור ביצועי המודלים, ותחזוקה אקטיבית לאורך זמן. מטרת שלב הפריסה אינה מסתכמת רק ביישום המודלים, אלא כוללת גם הבטחה שביצועיהם יישארו אופטימליים, מדויקים ורלוונטיים בסביבה הארגונית המשתנה. תיעוד ושיתוף התוצרים בפלטפורמות מקצועיות, כפי שפורט בפרק 1, הינם חלק בלתי נפרד מתהליך זה, שכן הם מאפשרים שקיפות, שיתוף פעולה והמשכיות.

**מבוא לניטור ותחזוקה: הקשר לנתוני עדכוני אבטחה של Microsoft**

קובץ הנתונים שברשותנו מתאר עדכוני אבטחה שהופצו עבור רכיבי תוכנה מגוונים במערכות ההפעלה של Microsoft. כל רשומה בקובץ מספקת מידע מקיף אודות עדכון ספציפי, ובכלל זה מזהה העדכון (Bulletin ID), קוד בסיס הידע (KB) הנלווה אליו, דרגת חומרת הפגיעות (Severity), השפעתה הפוטנציאלית (Impact), מערכות ההפעלה המושפעות (Affected Product) – לרבות Windows 7, 8.1, 10 וגרסאות שונות של Windows Server – וכן את רכיב התוכנה או החומרה הפגוע (Affected Component). בנוסף, הנתונים כוללים אינדיקציה האם העדכון מחייב הפעלה מחדש של המערכת (Reboot), וקישורים למזהי פגיעויות בינלאומיים (CVE). חשוב לציין כי רוב העדכונים מתייחסים לפגיעויות קריטיות כגון "Remote Code Execution" (ביצוע קוד מרחוק), אשר עלולות לאפשר לתוקפים להשיג גישה בלתי מורשית למערכת.

ניתוח מעמיק של נתונים אלו חיוני להבנת דפוסי עדכוני האבטחה של Microsoft ולזיהוי הפגיעויות המשמעותיות ביותר במערכות הפעלה וברכיבי תוכנה מגוונים. חשיבותו של ניתוח נתונים מסוג זה, ואיתו הצורך במערך מעקב ותחזוקה חזק, נובעת ממספר יתרונות אסטרטגיים ותפעוליים:

* **מניעת התקפות אבטחה:** פגיעויות קריטיות המפורטות בעדכונים, דוגמת "Remote Code Execution", מהוות סיכון משמעותי למערכות ולארגונים. ניתוח הנתונים מאפשר זיהוי מהיר של העדכונים בעלי החשיבות הגבוהה ביותר ותעדוף יישומם. לצורך זה, מערך מעקב ותחזוקה אפקטיבי הוא המפתח לצמצום חלון החשיפה לפגיעה.
* **שיפור מדיניות האבטחה הארגונית:** באמצעות ניתוח מתמיד של הפגיעויות ותגובה מהירה לתיקונן, ארגונים יכולים לשפר באופן אקטיבי את מדיניות האבטחה שלהם. גישה פרואקטיבית זו, המגובה במנגנוני ניטור ותחזוקה, מסייעת במניעת נזקים עתידיים, לרבות אובדן נתונים, פגיעה במוניטין והשלכות כספיות, הנובעים מפגיעויות בלתי מטופלות.
* **זיהוי מגמות ואיתור סיכונים מתפתחים:** ניתוח כמותי של הנתונים מאפשר לחשוף מגמות אבטחתיות מתפתחות, כגון עלייה בתדירות או בסוג הפגיעויות בתחום ספציפי לאורך זמן. תובנות אלו חיוניות למיקוד יעיל של משאבי אבטחה, לפיתוח אסטרטגיות הגנה ממוקדות, ולהיערכות לאיומים חדשים – וכל זאת מחייב מערכות ניטור מתמשכות.
* **תמיכה בדרישות תיעוד וציות (Compliance):** עבור חברות וארגונים גדולים, ניתוח ויישום מוסדר של עדכוני אבטחה הם חלק בלתי נפרד מדרישות רגולטוריות ומדיניות ציות פנימית. המידע המתקבל משמש כתיעוד מהימן וחיוני עבור ביקורות פנימיות ודוחות רגולטוריים, ומבטיח עמידה בתקני אבטחת מידע מחמירים – תוצר לוואי הכרחי של תהליך תחזוקה מוקפד.

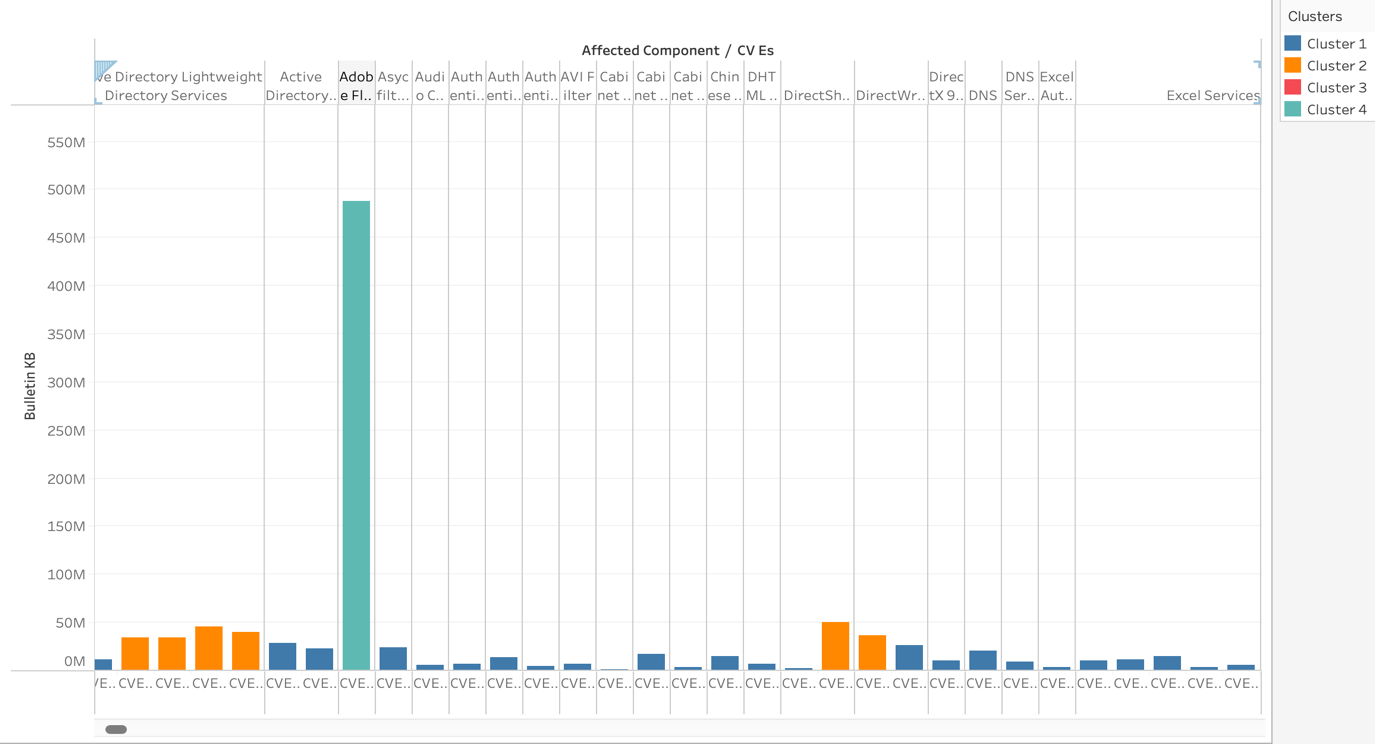
לסיכום, ניתוח זה מעניק למומחי אבטחת מידע תמונה בהירה, מפורטת ואקטואלית על מצב האבטחה של המערכות בארגון. הוא מהווה את הבסיס האנליטי עליו נבנה תכנון המעקב והתחזוקה בפרק זה, אשר חיוני לשמירה על סביבת מידע מוגנת ומעודכנת באופן שוטף.

## 1.2 אילו גורמים / השפעות צריכים להיות במעקב ? בדאטה שלנו יש הרבה מאוד לוגים שנרצה להיות במעקב אחריהם, כגון : Severity, ,Severity.1 Impact, CVEs ,Impact.1ועוד... נצרף את ניתוח מאפייני הנתונים שדיווחנו בדוחות הקודמים לשם סקירה וסימון התכונות עליהם נבצע את המעקב - כפי שניתן לראות, יש לנו הרבה תכונות קריטיות שהיינו רוצים לבצע עליהם מעקב מאחר וכל תכונה היא רלוונטית ויכולה להועיל לנו בתובנות ומציאת קשרים בהמשך לצורך חיזוי התקפות עתידיות. לצורך העניין אם יש לנו תכונה מסוג Affected Product או Affected Component כן היינו רוצים לדעת מה רמת הפגיעות שחוו או איזה רכיב או מוצר/שירות נפגע שכן אם נעקוב וננטר לוגים אלה נוכל להגיע למסקנה שאכן יש חולשה באותו שירות או רכיב. נוכל ללמד את המודל מתי, כמה ואיפה ע״פ מעקב וניטורם אילו מהי תקיפה וסיכון ממשי ומהו רישום לוג תקין במערכת. כלומר נרצה לחבר לאלגוריתם שלנו מערכת רישום (SIEM – Security Information and Event Management) מתוך אותה החברה שתהיה מעוניינת בכך, ובכך ללמד את המודל בצורה חיה את הרישומים השונים שנמצאים במערכת. נראה כמה גרפים על הדאטה שלנו מתוך Tableau אשר מראים כיצד היינו מנתחים ,וכיצד היינו שמים לב לאותם הלוגים הרשומים במערכת החברה.

## - ניתן לראות שימים 8 עד 14 הותקפו/ביצעו עדכונים לאורך כל החודשים. בנוסף ניתן לראות שאלו הימים לאורך החודשים שחוו הכי הרבה רישומי לוג חריגים כבין 2600 ל 3700 ( בניהם התקפות ועדכונים ).

## 

- בגרף זה ניתן לראות את כותרות הלוג מסודרות ע״פ חודשים לצד השפעות שבתוכם יש סיווג לרמת הקריטיות של אותה ההשפעה מנמוכה לקריטית. כלומר ע״פ גרף זה ניתן לדעת וללמד את המודל מהי בעיה קריטית ומהי בעיה נמוכה ולפי זה נוכל ללמד אותו לחזות עדכונים או מטרות שנרצה לממש ובכך למנוע התקפות עתידיות – בכך שנמנע חולשות. לצורך העניין בגרף זה ניתן לראות התקפה מסוג Elevation of Privilege (סוג תקיפה שבה התוקף מקבל הרשאות גבוהות יותר ממה שמותר לו) המסווגת כחשובה. בנוסף יש עוד המון התקפות שנוכל למצוא אשר רשומות בגרף זה כמו הידועה DDOS ו- Remote Code Execution.



- בגרף זה ניתן לראות שירות שנפגע באופן קשה מאוד, המיוחס לו ככמעט חצי מיליון מידעים שנפגעו מאותו שירות זה, השירות הוא Adobe Flash Player, אשר היה תוסף לדפדפים השונים והיה יעד אסטרטגי לאקרים בשל פרצות רבות לאותו השירות.

## 2.2 כיצד נמדוד וננטר את תוצאות ודיוק המודלים ?

עבור כל ממצא נציין את הרכיבים ותכונות המעקב כמו שווי שוק או עונתיות. כיצד נמדוד ואיך ננר התקפות ודיוק מודלים ונקבע האם יש מצב בו תוקפו של המודל פג או שינויים צפויים של הנתונים למשל האם המודל שבנינו יהיה רלוונטי לאורך זמן – ימים , חודשים , שנים ? – מה טווח זמן הרלוונטיות של המודל ?

למדידה וניטור של תוצאות ודיוק מודלים, נשתמש במדדים כמותיים איכותיים בהתאם לסוג המודל .  
בנוסף נשלב כלי ניטור כדי לוודא שהתוצאות נשמרות גם לאורך זמן.

כדי למדוד את תוצאות דיוק המודלים נשתמש במדדים כמותיים:

דיוק המודל – אחוז התחזיות הנכונות מכלל התחזיות.

Precision ו-Recall – מדדים לזיהוי של נתונים לא מאוזנים.

F1 Score – משלב את Precision ו-Recall למדד אחד מאוזן.

Confusion Matrix – טבלת שמשקפת את סוגי הטעויות שהמודל מבצע, עורך השוואה בין תוצאות המודל לתוצאות האמיתיות של הנתונים.

עבור כל מדד נציג את היתרונות , החסרונות ואיך נמדוד בצורה אופטימלית.

|  |  |  |  |
| --- | --- | --- | --- |
|  | יתרון | חסרון | מדידה אופטימלית |
| Accuracy | טוב לנתונים עם מחלקות מאוזנות | לא מבחין בין סוגי הטעויות | ניקוי נתונים ומדדים של חוסר איזון. |
| Precision | חשוב כאשר שגיאת המודל גבוהה | חוסר איזון בניבוי אופטימי של המודל | איזון עם Recall ושימוש בפרמטרים בצורה נכונה של המודל |
| Recall | חשוב למניעת שגיאות false-negatives | עלול להוביל לעליה בשגיאת הניבוי | שילוב עם Pricision וF1, שימוש בנתונים מגוונים. |
| F1 Score | ממוצע הרמוני מאוזן בין Precision וRecall | אין פירוט מלא על הטעויות. | שילוב מדדים נוספים, ומניעת הטייה. |

כדי לנטר את התוצאות של המודלים נשתמש בשיטות שונות:

|  |  |  |  |
| --- | --- | --- | --- |
|  | יתרון | חסרון | מדידה אופטימלית |
| מעקב אחרי מגמות ביצועים לאורך זמן | זיהוי ירידה מוקדמת בביצועים והתאמת המודל בזמן | דורש מעקב רציף | השוואת ביצועים על נתונים חדשים ומדידה תקופתית. |
| **ויזואליזציה של תוצאות** | הצגת מידע ברור ואינטואיטיבי על ביצועי המודל בDashboard | יכול להסתמך על פרשנות אישית | יצירת Dashboards מעודכנים בזמן אמת וסקירתם קבועה |
| ניטור שינויים בנתונים | מניעה ירידה בביצועים בעקבות שינוי | זיהוי מורכב ודורש משאבים | שימוש בכלים סטטיסטיים לזיהוי דיפרנציאציה בנתונים ומתן התראה אוטומטית |
| ניסויים מבוקרים | בחירה מותאמת בין גרסאות | דורש משאבים וזמן , לא תמיד ניתן לביצוע | ניסוי מבוקר ובחינת ביצועים סטטיסטיים |
| איסוף ומשוב ממשתמשים | משפר התאמה לצרכי המשתמש ומעלה את איכות המודל | לא מייצג את כלל המשתמשים , יכול להיות סובייקטיבי | איסוף משוב באופן שיטתי וקבוע , |
| **התראות** | מאפשר תגובה מהירה לבעיות ומונע נזקים | סיכון להתראות שגויות | הגדרת סף רגישות מתאים והתאמת מערכת ההתראות לאורך זמן |

## 3.2 כיצד נקבע מתי פג התוקף של כל דגם ?

קביעת פג תוקפו של המודל מתבצעת על בסיס שילוב של ניטור ביצועים, זיהוי שינויים בנתונים, והערכה של תקופת זמן.

**מעקב ביצועים לאורך זמן** - נבדוק את ביצועי המודל על **נתונים חדשים** לאורך זמן.

אם המדדים יורדים מתחת לסף שנקבע , אזי המודל כבר פג וצריך לעדכן אותו או להחליפו.

**זיהוי שינויים בנתונים** - אם התפלגות התכונות שהמודל מקבל משתנה , ביצועיו של המודל יפחתו.

שינויים אלו יכולים להתרחש בעקבות , טרנדים , זמן או שינויים חיצוניים.

גילוי שינויים באמצעות ML יכול להפעיל התראה על פג תוקף.

הגדרת תקופת תוקף מראש - נוכל להגדיר מראש טווח זמן בו המודל יהיה רלוונטי לדגם, כמו:

שנים – במודלים שפועלים בסביבה יציבה מאוד יכול להאריך את זמן הרלוונטיות אך חשוב לדעת מתי פג תוקפו.

חודשים – מודלים של שווקים משתנים או בסביבות דינמיות.

תמונה שמכילה טקסט, צילום מסך, עיצוב, עיצוב גרפי

תוכן בינה מלאכותית גנרטיבית עשוי להיות שגוי.טרנדים – תקופת התוקף עשויה להיות קצרה מאוד ,בהתאם לעלייה וירידה של טרנדים.