

# **Secure IDE (IDE למפתחי נוזקה):**

***מסמך אפיון***

**אדם ליבנה 209621325**

**גרסה 2.0**

**תיכון הראל, מבשרת ציון**

**הנדסת תוכנה 883589**

**חנן רוזנטל**

**14/10/2019**

## היסטוריית גרסאות המסמך

|  |  |  |
| --- | --- | --- |
| **תאריך** | **גרסה** | **תקציר השינויים** |
| 14/10/2019 | 1.0 | טיוטה ראשונית |
| 22/03/2020 | 2.0 | הוספה של תוכן עניינים ושינויים בכתיבה |
|  |  |  |
|  |  |  |

## **תוכן עניינים**

[**Secure IDE (IDE למפתחי נוזקה):** 1](#_Toc35796251)

[היסטוריית גרסאות המסמך 1](#_Toc35796252)

[**תוכן עניינים** 2](#_Toc35796253)

[**1. הקדמה:** 2](#_Toc35796254)

[**1.1 מטרה** 3](#_Toc35796255)

[**1.2 המוצר** 3](#_Toc35796256)

[**1.3 הגדרות** 3](#_Toc35796257)

[**1.4 תקציר** 4](#_Toc35796258)

[**2. תיאור כללי:** 4](#_Toc35796259)

[**2.1 פונקציונליות** 5](#_Toc35796260)

[**2.2 קהל היעד** 5](#_Toc35796261)

[**2.3 אילוצים עיקריים** 5](#_Toc35796262)

[**2.4 הנחות ותלויות** 6](#_Toc35796263)

[**2.5 סיקור מצב השוק כיום (אופציונלי)** 6](#_Toc35796264)

[**3. דרישות מפורטות:** 7](#_Toc35796265)

[**3.1 דרישות פונקציונליות** 7](#_Toc35796266)

[**3.2 דרישות של ממשקים חיצוניים** 8](#_Toc35796267)

[3.2.1 ממשקי משתמש (GUI): 8](#_Toc35796268)

[3.2.2 ממשקי חומרה: 8](#_Toc35796269)

[3.2.3 ממשקי תוכנה: 8](#_Toc35796270)

[3.2.4 ממשקי תקשורת: 8](#_Toc35796271)

[**3.3 דרישות לא פונקציונליות** 8](#_Toc35796272)

[3.3.1 דרישות ביצועים: 9](#_Toc35796273)

[3.3.2 דרישות מהימנות: 9](#_Toc35796274)

[3.3.3 דרישות זמינות: 9](#_Toc35796275)

[3.3.4 דרישות אבטחה: 9](#_Toc35796276)

[3.3.5 דרישות תחזוקה: 9](#_Toc35796277)

[3.3.6 דרישות ניידות: 9](#_Toc35796278)

[**3.4 דרישות בסיס נתונים** 9](#_Toc35796279)

[**3.5 דרישות נוספות** 9](#_Toc35796280)

[**4. נספחים:** 10](#_Toc35796281)

[**4.1 ניהול הפרויקט** 10](#_Toc35796282)

# **1. הקדמה:**

החלק הזה ישמש להצגה כללית של מסמך האפיון - הוא נועד לאפשר לקורא להבין במה מדובר - מה זה בכלל הפרויקט הזה ומה מופיע בהמשך המסמך.

מסמך זה מתאר את המבנה של הפרויקט המסכם של מגמת סייבר בתיכון הראל במסגרת תוכנית "גבהים" של משרד החינוך. במסמך מופיע תיאור של המוצר, קהל היעד, רכיבים מרכזיים ועוד.

## **1.1 מטרה**

מה המטרה של המסמך ומי קהל היעד שלו

מטרת הפרויקט היא ליצור סביבת פיתוח תוכנה המהווה ערכת למידה ועבודה בתחום בדיקת בעיות אבטחה של מערכות מחשוב, כאשר קהל היעד של הפרויקט הוא מתכנתים אשר רוצים ללמוד ולעסוק בתחום אבטחת מחשבים.

## **1.2 המוצר**

חלק זה צריך להכיל את הדברים הבאים: - שם המוצר אותו מפתחים (ובאמצעותו נתייחס למוצר בהמשך המסמך) - הסבר כללי של מה המוצר עושה (ואם צריך- מה הוא לא עושה) - תיאור של מטרות המוצר ויתרונותיו העיקריים. נסו להיות מדויקים וספציפיים כמה שניתן.

שם המוצר: secureIDE

המוצר מספק סביבה תכנותית – לימודית בתחום אבטחת מחשבים אשר בעתיד תתמוך במספר שפות תכנות, כמו כן גם ספרייה עם חומרי לימוד ומידע מועיל, ומכונה ווירטואלית (Sandbox) למטרות בדיקה של תוכנות זדוניות ומערכות מידע.

מטרת המוצר לספק כמות מקסימלית של כלים ללמידה ופיתוח תוכנות בתחום אבטחת המחשבים. יתרונות המוצר בכך שהוא מרכז את הכל תחת תוכנה אחת, ושהוא מתרכז בעיקר בתחום אחד.

בנוסף, ישנו גם אתר שממנו אפשר להוריד את התוכנה וגם סטאפ וויזארד.

## **1.3 הגדרות**

בחלק זה יש לציין את כל ההגדרות ,המושגים וראשי התיבות בהם תשתמשו בהמשך המסמך.

* IDE (Integrated Development Environment) - סביבת פיתוח משולבת, [תוכנת מחשב](https://he.wikipedia.org/wiki/%D7%AA%D7%95%D7%9B%D7%A0%D7%94) המסייעת ל[מתכנתים](https://he.wikipedia.org/wiki/%D7%9E%D7%AA%D7%9B%D7%A0%D7%AA) לפתח תוכנה.
* אבטחת מחשבים – תחום רחב הכולל בתוכו כל נושא הקשור בגרימת/מניעת נזק לתפקוד של מערכת ממוחשבת.
* ספריות - אוסף של תת-תוכניות המשמשות לפיתוח [תוכנה](https://he.wikipedia.org/wiki/%D7%AA%D7%95%D7%9B%D7%A0%D7%94). ספריות מכילות קוד או מידע, שמספקים שירות לתוכניות עצמאיות.
* GUI (graphical user interface) – ממשק משתמש גרפי, צורה אינטואיטיבית ופשוטה לאינטראקציה ששל המשתמש עם המחשב בעזרת גרפיקה.
* UI (user interface) – ממשק משתמש, הדרך בה המשתמש מתקשר עם התוכנה הרצה על המחשב.
* API (Application Programming Interface) - ממשק תכנות יישומים, ערכה של ספריות קוד, [פקודות](https://he.wikipedia.org/wiki/%D7%A4%D7%A7%D7%95%D7%93%D7%94_(%D7%9E%D7%97%D7%A9%D7%91)), [פונקציות](https://he.wikipedia.org/wiki/%D7%A4%D7%95%D7%A0%D7%A7%D7%A6%D7%99%D7%94) ו[פרוצדורות](https://he.wikipedia.org/wiki/%D7%A9%D7%92%D7%A8%D7%94) מן המוכן, בהן יכולים ה[מתכנתים](https://he.wikipedia.org/wiki/%D7%9E%D7%AA%D7%9B%D7%A0%D7%AA) לעשות שימוש פשוט, בלי להידרש לכתוב אותן בעצמם כדי שיוכלו להשתמש במידע של היישום שממנו הם רוצים להשתמש לטובת היישום שלהם.
* VM (virtual machine) - מכונה וירטואלית, [תוכנה](https://he.wikipedia.org/wiki/%D7%AA%D7%95%D7%9B%D7%A0%D7%94) היוצרת סביבה הנחוצה להפעלתה של תוכנה אחרת, מבלי שיהיה צורך במימוש פיזי של סביבה זו. מכונה וירטואלית משחררת את התוכנה המתבססת עליה מתלות ב[חומרה](https://he.wikipedia.org/wiki/%D7%97%D7%95%D7%9E%D7%A8%D7%94) מסוימת, משום שכל חומרה שעליה תמומש המכונה הווירטואלית תאפשר את הפעלת התוכנה. בנוסף, המכונה הווירטואלית מדמה מערכת הפעלה, מה שמאפשר לערוך בדיקות העלולות להיות הרסניות מבלי לדאוג לשלמות המחשב המריץ את הבדיקה.

## **1.4 תקציר**

הסבר מה מופיע בהמשך המסמך ופירוט כיצד זה מסודר לאורך המסמך

במסמך זה מופיעה סקירה מקיפה של הדרישות מהפרויקט, הדרך בה הפרויקט אמור להיראות ודרישות ונושאים נוספים הנודעים לעבודה על הפרויקט.

# **2. תיאור כללי:**

החלק הזה מתאר את הגורמים העיקריים המשפיעים על המוצר ועל דרישותיו. חלק הזה לא מפרט דרישות ספציפיות אלא רק עוזר להבין את הדרישות.

על המוצר משפיעה מהירות הפעולה של התוכנה, פשטות ובהירות של ממשק המשתמש, אמינות של הספריות והפעולות, גמישות של כל המבנה

## **2.1 פונקציונליות**

חלק זה יכיל סיכום של הפונקציות העיקריות במערכת (לא פונקציות של שפת תכנות, אלא באופן מילולי - מה הדברים שהמערכת עושה)

המערכת מספקת סביבה תכנותית – לימודית בתחום אבטחת מחשבים אשר תומכת במספר שפות תכנות (עם אפשרות להוסיף עוד), כמו כן גם ספרייה עם חומרי לימוד ומידע מועיל (עם אפשרות להוסיף עוד), מכונה ווירטואלית למטרת בדיקה, משימות (כמו באתר hackthissite), אפשרות להרים שרת אשר דרכו ניתן יהיה להמשיך את העבודה מכל מכשיר שהוא, ותמיכה מלאה לכל מערכות ההפעלה (גם לטלפונים).

## **2.2 קהל היעד**

תיאור כללי של קהל היעד של המערכת - מה תפקידם ,ניסיונם, יכולתם ללמוד את ממשק המערכת וכל דבר אחר שישפיע על דרישות המערכת. יתכן יותר מפרופיל אחד של משתמש במערכת, ולכן יש לפרט את כל הסוגים וההשלכות/משמעויות שלהם מבחינת אופי ורקע טכנולוגי או פונקציונלי. ("מי הולך להשתמש במערכת?")

3 קהלי יעד: מתכנתים בעלי רקע שצריכים סביבת עבודה ברמה גבוהה, מתכנתים בעלי רקע שלא מכירים את תחום האבטחה, ואנשים שרק התחילו לתכנת ומעוניינים ללמוד על תחום האבטחה.

## **2.3 אילוצים עיקריים**

הגבלות ואילוצים עיקריים שצריך לקחת בחשבון בזמן עיצוב המערכת - אילוצי חומרה, סביבת ריצה, זמן ריצה וכו'.. "המערכת דורשת חיבור לאינטרנט" / "המערכת צריכה לתמוך במספר רב של מערכות הפעלה" / "המערכת צריכה –(למשל לתמוך במספר משתמשים רב")

המערכת צריכה:

- לתמוך במספר רב של מערכות הפעלה

- לתמוך במספר רב של שפות תכנות (כולל אסמבלי וC)

- לאפשר עבודה רציפה על מספר מכשירים

- לא לתפוס יותר מדיי זיכרון

- להתחבר לאינטרנט, אבל גם לעבוד בלי חיבור

- לאפשר הגדלת/הקטנת/שינוי פונט וצבע הטקסט והמראה

- להציע דבאגינג

- להציע תיקון שגיאות (תכנותיות/תחביריות)

- להציע אילו ספריות לכלול, להראות אילו ספריות קיימות ומה קיים בהן

- לאפשר גישה לקוד המקור של הספריות במידת האפשר

- להציע תיעוד אוטומטי

- שיהיה dark mode

-

## **2.4 הנחות ותלויות**

כל הנחה שיש לנו לגבי המערכת וסביבת הריצה שלה, ששינוי שלה עשוי לגרום לשינוי באפיון ובעיצוב - לדוגמא "המערכת תרוץ בסביבת לינוקס" זו הנחה המשפיעה על אפיון ועיצוב המערכת. אם נשנה את זה בהמשך נצטרך לעבור שוב על המסמך ולשנות את האפיון והעיצוב של המערכת בהתאם (זה שונה מהסעיף הקודם- אילוצים אינם משתנים, זהם משהו שנקבע בתחילת העבודה ומשפיע רק על האפיון הראשוני). אל תתנו להנחות הללו להגביל אתכם בהמשך (אם תגלו שלמשל עשו זאת). – עדיף בסוף להריץ את השרת המרכזי מעל וינדוס

* המשתמש מגיע עם איזשהו ידע מוקדם במחשבים
* המערכת תרוץ על מחשב עם מערכת הפעלה מסוג windows
* אין צורך לעדכן את התוכנה ואת רכיביה
* למחשב ולמשתמש יש גישה לאינטרנט
* למחשב יש לפחות כ 700 MB פנויים בזיכרון

## **2.5 סיקור מצב השוק כיום (אופציונלי)**

במידה וידוע לכם על מוצרים קיימים כיום (אתרים, תוכנות, אפליקציות וכו') שנותנים פתרון דומה (גם אם חלקי) לרעיון שלכם - קשרו אליהם כאן או תארו אותם בקצרה. התמקדו באופן שבו הם פותרים או מתייחסים לסוגיות דומות לאלו שבחרתם, מה הפיצ'רים העיקריים שהם מציעים ובמה הם שונים מהמוצר שלכם.

כיום בשוק קיימים כל מני IDEs כמו Eclipse , PyCharm , NetBeans , Android Studio , VS. השניים העיקריים אשר בהם רוב משתמשי הג'אווה מתכנתים הם אקליפס ונטבינס. שני IDE אלה מאפשרים תכונות רבות ומגוונות. ביניהן, שימוש בקונסולה ובאזור קוד וכן מצב כהה ובהיר לתוכנה, תכונות אשר מופיעות גם בפרויקט שלי. אך למרות הדמיון, ישנן מספר תכונות אשר מבדילות את הפרויקט שלי ומהוות את עיקר החידוש שבו. כיום אין בשוק IDE אשר משלב כחלק בסיסי ואינטגרלי בתוכו הרצת תוכנות בתוך Sandbox כתכונה המובנית כברירת מחדל. בנוסף, כיום גם אף IDE אינו נותן אפשרות לשילוב של חומרי לימוד כגון כל מיני tutorials באתרים וסרטוני הסברה למיניהם, וחומרים משפטיים כחלק מהתוכנית עצמה.

# **3. דרישות מפורטות:**

זהו החלק העיקרי והחשוב ביותר של מסמך האפיון. פרק זה יכיל את הדרישות המפורטות מהמערכת שינחו את מעצבי המערכת, המפתחים, והבודקים בהמשך. כל דרישה שתפורט בחלק זה צריכה להיות:

* מזוהה על ידי שם (או עדיף - מספר) באופן ייחודי
* מתועדפת (כלומר מה העדיפות שלה כלפי הדרישות האחרות)
* ניתנת לבדיקה
* נכונה ושלמה
* עקבית ולא מתנגשת עם דרישות אחרות

שימו לב שישנם מספר סוגים של דרישות, כפי שמפורט בכותרות כאן:

יש להשקיע בחלק זה ולפרט את הדרישות בצורה ברורה ונוחה להבנה. יש לזכור שזה לא מסמך עיצוב- יש להמנע מתיאור שפת תכנות) שיגיע בשלב מאוחר יותר של עבודתכם. – מפורט מדי שמגביל את העיצוב הטכני (למשל את הדרישות רצוי לחלק לקטגוריות הבאות:

## **3.1 דרישות פונקציונליות**

תיאור של פיצ'רים ספציפיים במערכת. לכל דרישה פונקציונלית מומלץ לפרט:

* הסבר כללי
* קלט
* תהליך
* פלט
* טיפול בשגיאות.

בכל "נקודת זמן" של תיאור use case)( ניתן (ואף רצוי) להציג את הדרישות הפונקציונליות בפורמט של תרחיש שימוש התרחיש יש לפרט את כל האפשרויות השונות שעומדות בפני המשתמש או המערכת, ומה יכול לקרות בכל התפצלות שכזו (שימו לב להתייחס גם למקרים שליליים, כמו שגיאות או נתונים לא מתאימים, ולא להניח שהמשתמש תמיד פועל בדרך שלה אנו מצפים)

IDE:

* החלק העיקרי של הפרוייקט אותו מפעיל המשתמש בתחילת השימוש. מספק GUI שדרכו מופעלות כל שאר התוכנות.
* התוכנית רצה כשלוחצים על האייקון שבשולחן העבודה/מסך הבית.
* .הקלט הוא בעיקר דרך לחיצות על כפתורים וכתיבה לתוך תיבות טקסט, הפלט הוא הצגה של מידע עבור המשתמש על המסך.
* המשתמש יוריד את התוכנה מאתר באינטרנט, ודרך הGUI יפעיל את כל הפונקציונליות שמספק הפרויקט
* עבור טיפול בשגיאות יהיה חלק של מידע באתר האינטרנט.

VM:

* מופעל ע"י לחיצה על כפתור בIDE
* ניתן לעשות שימוש בתוכנת virtualBox

## **3.2 דרישות של ממשקים חיצוניים**

### 3.2.1 ממשקי משתמש (GUI):

צריך שהממשק יהיה פשוט להבנה ולתפעול

דרושה תמיכה בעבודה עם מספר מסכים במקביל

תמיכה ביישור טקסט לפי השפה

### 3.2.2 ממשקי חומרה:

תלוי מכשיר ומערכת הפעלה

### 3.2.3 ממשקי תוכנה:

אפשרות לתקשורת בין שפות תכנות שונות, בין אם דרך שרת, מסמך משותף, קריאות מערכת להפעלת קובץ הרצה, או תמיכה של השפה דרך ספרייה או תמיכה מובנית.

### 3.2.4 ממשקי תקשורת:

חיבור לדפדפן ולאינטרנט

הפעלה של שרת ולקוח

## **3.3 דרישות לא פונקציונליות**

דרישות מהמערכת שלא מתבטאות בפיצ'ר ספציפי או בתהליך ספציפי שמתרחש במערכת אבל משפיעות על אופן עיצובה ומימושה, לדוגמא:

### 3.3.1 דרישות ביצועים:

המערכת צריכה לעבוד בקצב טוב ללא תקלות

### 3.3.2 דרישות מהימנות:

אסור שהתוכנה תקרוס

### 3.3.3 דרישות זמינות:

צריכה להיות זמינה תמיד, גם תוך כדי עדכון

### 3.3.4 דרישות אבטחה:

אסור שיהיו פרצות וback doors

### 3.3.5 דרישות תחזוקה:

מינימליות, עם אפשרות להוציא עדכונים

### 3.3.6 דרישות ניידות:

צריכה להיות קרוס – פלטפורם

## **3.4 דרישות בסיס נתונים**

איזה נתונים יישמרו ● / קובץ . ניתן לציין גם כמה אופציות) DB( באיזה פורמט יש לשמור את המידע ● מה יכולות האחסון הנדרשות (נפח אחסון לדוגמא) ● מה רמת האבטחה הנדרשת ●

חייב להיות נפח אחסון בגודל של התוכנית, אין צורך באבטחה של הזיכרון, יישמרו הפרויקטים שנכתבו, קבצים של התוכנית עצמה, וקבצי טקסט שמכילים הסברים ומידע (כולל מסמך זה)

פורמטים נחוצים:

- קובצי שפות (.java, .asm, .py)

- קובצי ספריות (.dll, .elf)

- קובצי הרצה (.exe, .elf)

- קובצי טקסט (.txt)

- לווינדוס: batch files (.bat, .cmd)

## **3.5 דרישות נוספות**

כאלה שלא ידעתם לאיזה חלק הם שייכות

# **4. נספחים:**

## **4.1 ניהול הפרויקט**

|  |  |
| --- | --- |
| יעד (מה יש לבצע) | זמן ביצוע (מתאריך עד תאריך) |
| לסיים את הסקיצה של המסמך תכנון | עד אחרי סוכות |
| להגיש את כל המסמכים והתוכנה | עד אחרי פסח |
|  |  |