

**פרויקט גמר הגנת הסייבר - FinalWall**

**נושא העבודה: חומת אש אפליקטיבית - WAF**

**שם תלמיד:** דניאל ספוז'ניקוב

**ת.ז תלמיד:** 215673369

**שם בית ספר ועיר :** קריית החינוך ע"ש עמוס דה-שליט, רחובות

**שם המנחה :** הילה גורן ברנע

**מועד הגשה :** 29/05/2024

תוכן עניינים

[1. מבוא 3](#_Toc121815660)

[2. תיאוריה 5](#_Toc121815661)

[3. תוצר סופי 10](#_Toc121815662)

[4. תהליך כתיבת הפרויקט 24](#_Toc121815663)

[5. מרכיבי פתרון 25](#_Toc121815664)

[6. תסריטי בדיקה 33](#_Toc121815665)

[7. רפלקציה 34](#_Toc121815666)

[8. הוראות התקנה ותפעול 35](#_Toc121815667)

[9. ביבליוגרפיה 35](#_Toc121815668)

[10. נספחים 36](#_Toc121815669)

1. מבוא
   1. נושא העבודה  
      הפרויקט שאני בחרתי ליישם במסגרת לימודי במגמת סייבר – תכנון, ופיתוח מערכות מידע הוא – Web application Firewall או כשמו המקוצר – WAF. את הפרויקט שלי ניתן למצוא בפלטפורמת Github אותו אני מכנה – FinalWall. מערכות WAF מהוות כחלק מרכזי במערך ההגנה על תוכנות Web אך הן לא היחידות, הן מצטרפות למגוון רחב של כלים אשר המטרה שהמשותפת שלהם היא להגן על האתר עליו הופקדו. תפקידה העיקרי של חומת אש אפליקטיבית היא לתחקר את התעבורה העוברת דרכה, לנטר אותה ולהציג אותה בדרך ידידותית. מערכת FinalWall מקנה ללקוח מגוון כלים לפיקוח התעבורה על תוכנת Web יחידה. כאשר ממוקמת בין לקוח לשרת, FinalWall יודעת לחקור את התעבורה אותה היא מקבלת ולנטר אותה בצורה יעילה ומהירה תוך שימוש במספר שיטות לזיהוי פקטות (חבילות מידע ברשת) חשובות.
   2. מטרות מרכזיות

המטרות היישומיות של הפרויקט הן:

* פיתוח מערכת יעודית אשר יודעת לא רק למצוא נוזקות ומתקפות ברשת אלא גם לדווח אותן בצורה יעילה וידידותית.
* ניתוח הודעות HTTP משני צדדי התקשורת (לקוח ושרת) וממשפחות שונות של מתקפות.
* ממש משתמש מתקדם מבוסס Web בעל 2 מערכות – Backend מבוסס FLASK ו-Frontend מבוסס Vue.js.
* פיתוח Package בפייתון אשר מאפשרת לכל מפתח ברחבי העולם ליצור WAF לוקלי או Remote.

המטרות האישיות של העבודה הן:

* למידה של מערכת ה-WAF לפרטי פרטים תוך שילוב חקירת מתקפות ברשת ויישום קוד אשר יודע לבלום אותן.
* ללמוד לפתח GUI מבוסס Web אשר יהיה מתקדם ויאפשר ללקוח למצות את יכולותיה של FinalWall בצורה מירבית.
* פיתוח פרויקט גדול ורציני הדורש עבודה המשכית ב- Github ובסטנדרטים הגבוהים של המגמה.
  1. רציונל  
     לאדם שאוהב פיתוח ואת כל מה שקשור לעולמות הרשת, חומת אש אפליקטיבית היא שילוב מדהים בין השניים. מצד אחד, יש בה את כל מה שאני אוהב ברשת – מחקר בנושא מתקפות ב-Web. יישום טכניקות הגנה מתקדמות, פיתוח פרופילים על לקוחות – Social Engineering. ומצד שני, פיתוח. תמיד אהבתי לפתח תוכנות ובדגש על רשתות ולכן בחרתי בפרויקט זה, מכיוון שהוא משלב את כל תחומי העניין שלי בצורה המיטבית ביותר. כמו כן, יש בפרויקט זה נושא מאוד חשוב והוא HTTP. מכיוון שמדובר בפרוטוקול כל כך חשוב ושיש לו חשיבות עצומה בעולם המודרני, רציתי ללמוד עליו הרבה יותר. לדעתי את כל הניואנסים שלו כשזה בא ל-Headers, Encryption ולחקור טכניקות WAF BYPASS שונות כמו הסתרת מידע עם שלבי Encapsulation שונים וכו. ניתן לומר שהמניע העיקרי בפרויקט הוא האהבה שלי לפיתוח מערכות סייבר מתקדמות הקשורות לווב.
  2. קישור לחומר הנלמד

הפרויקט מקושר לחומר הנלמד בכיתות י"א וי"ב במספר דרכים ונושאים.  
תחילה, במובן התקשורתי – המערכת בנויה על מספר שרתים בינהם שרת Frontend, Backend, שרת Tunneling אשר מאפשר תקשורת רציפה עם כל Cluster ואת השרת Reverse Proxy עצמו של כל Cluster. כל שרת למעט שרת ה-Frontend, הוא שרת אסינכרוני אשר בנוי על Low level API של סוקטים בספרייה Asyncio של פייתון. כל Cluster עצמו מנהל תקשורת עם הלקוח והשרת Web כמו כן, עם הHIGH LEVEL API של Asyncio. – Streams. כל Stream בעצם עוטף את הSocket של מערכת ההפעלה ומאפשרת שימוש קל יותר בספרייה ובכך לממש מערכת תקשורת טובה יותר.   
  
בנוסף, בהיבט מערכות ההפעלה, FinalWall הינה מתעסקת בהיבטים שהם Low level אך היא מתפעלת שרתים ולכן בהינתן Connection מסוים כל שרת יפתח Coroutine אשר ינהל את התקשורת עם אותו Endpoint. ניתן להשוות Coroutine ל- Thread בכך שהוא משעה ומחדש את ביצוע המשימות שלו בסדר מסוים וכך מאפשר מקביליות בין המשימות עליו הוא אחרי.

1. תיאוריה
   1. תיאוריה
   2. מוצרים קיימים

לפרט על כמה מהמוצרים הקיימים בשוק, כמה שורות על כל מוצר. לסכם בטבלה:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | הפרויקט שלי | מוצר קיים א' | מוצר קיים ב' | מוצר קיים ג' |
| קריטריון השוואה א' | V | V |  | V |
| קריטריון השוואה ב' | V | V | V | V |
| קריטריון השוואה ג'... (להוסיף כמובן עוד קריטריונים) | V |  | V | V |

1. תוצר סופי
   1. תיאור הפרויקט

כמה שורות תיאור...

רכיבי המערכת העיקריים:

* פירוט
* על
* כל
* רכיב
  1. אלגוריתמים עיקריים
  2. דרישות ומגבלות מערכת

למערכת מספר דרישות ומגבלות.

**בהיבט הדרישות:**

* לפרט על דרישות כמו שלמות, מהירות, תמיכה בסוגי פורמטים שונים...

בהיבט המגבלות:

* היבטים כמו תמיכה רק במערכות הפעלה מסוימות, רק בכמות משתמשים מסוימת וכו'. לפרט על כל מגבלה.
  1. התייחסות לנושא אבטחה
  2. ממשק משתמש

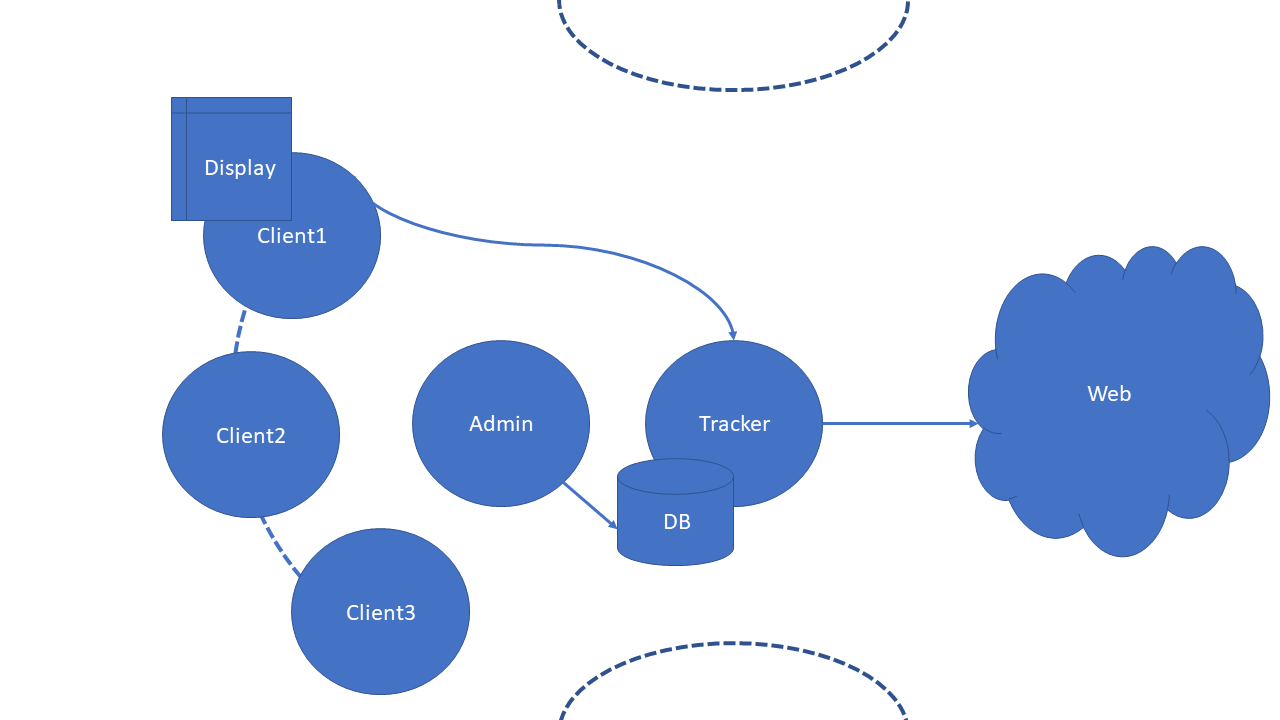
צילום תרשימי מסך של כל אחד מהחלונות לפי הלוגיקה והסדר של התהליך בפרויקט. הסבר קצר על כל חלון.

* 1. תרחישים עיקריים

תרשים זרימה לכל אחד מהתהליכים

1. תהליך כתיבת הפרויקט
   1. תהליך הפרויקט
   2. אתגרים ואופציות שונות למימוש
2. מרכיבי פתרון
   1. תיחום הפרויקט

* תקשורת –
* מערכות הפעלה –
* תצוגה –
* אבטחה –
* מבנה נתונים –
* שימוש במסד נתונים
* ארכיטקטורת קוד –
* תיעוד –
  1. סביבת העבודה (טכנולוגיה)
* שפות התכנות:
* סביבת פיתוח:
  1. מבט טופולוגי

המערכת מורכבת משרת ראשי, מ-Clients ומ-Admin.

Client1, Client2, Client3 מייצגים עמיתים ברשת הפנימית, שמשתפים pieces זה עם זה.

Client1 ראה שחסר לו piece ושהוא איננו נמצא ברשת הפנימית, לכן הוא פונה לרשת החיצונית לצורך השגת ה-piece החסר.

כל העת ה-Admin ניגש למסד הנתונים המשותף לו ולשרת הניהול ומעדכן את הסטטיסטיקות.

* 1. מבנה נתונים

:Tracker

* admin\_ips – רשימה של כל ה-admins המחוברים
* not\_listening – רשימה של העמיתים אליהם לא יאזין שרת הניהול, עמיתים יוספו למבנה נתונים זה כשהם מעלים קובץ לשרת הניהול ויוסרו בסיום ההעלאה
* requests – רשימה של כמות הפניות לשרת הניהול ומילון של המשתמשים שפנו וכמה פניות ביצעו

User:

* Pieces – מילון של ה-pieces ואילו עמיתים מחזיקים ב-pieces אלו
* peer\_list – רשימה של האובייקטים של העמיתים השונים (לצורך הורדה מהם)
* peer\_thread – מילון של ה-threads הפתוחים ואובייקט העמית אליו ממנו מורד piece ב-thread זה
* currently\_connected – עמיתים אליהם מחובר משתמש המערכת בעת ההורדה
* peers – רשימה של כל העמיתים שהושגו מהשרתי הניהול השונים ברשת
* announce\_list – שרתי הניהול הזמינים (במקרה של קובץ גלובלי) או העמיתים הזמינים (במקרה של קובץ מקומי)

Admin:

* ip\_files – רשימה שמכילה בתוכה את כל העמיתים שלא שיתפו בטווח זמן נתון מרגע שנוספו, שולחת לשרת הניהול הודעה להסרתם ומאפסת את הרשימה
  1. מסד נתונים
  2. מבט מודולרי
  3. פירוט מודלים עיקריים

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Input\Output | Function | Class |
|  |  |  |  |
|  |  |  |  |

1. תסריטי בדיקה
   1. דגשים בבדיקה
   2. תסריטי בדיקה עיקריים
2. רפלקציה
   1. לוח זמנים מוערך לניהול הפרויקט:
   2. אתגרים ותרומה אישית
   3. תובנות
3. הוראות התקנה ותפעול
   1. תצורה ודרישות קדם
   2. התקנה
4. ביבליוגרפיה

במהלך כתיבת הפרויקט הסתמכתי על מספר מקורות מידע:

1. נספחים

לינק לגיטהאב