פרויקט בהנדסת תוכנה

תכנון ותכנות מערכות בהתמחות הגנת סייבר  
סמל מקצוע: 883589

**Eagle Eye**

מערכת לניטור ובקרת תקשורת

שם התלמיד: אופק ארז

מספר זהות: 214273393

שם בית ספר: קריית חינוך אמירים

עיר: ראשון לציון

שם המנחה: מוטי מתתיהו

מועד הגשת המסמך:

**תוכן עניינים וראשי פרקים**

# פרק א – Eagle - Eye Project - ייזום

1. תיאור ראשוני של המערכת

מבוא

בחרתי לפתח תוכנה לניטור תעבורת רשת מפני שכאשר השתמשתי בתוכנה הקיימת לניטור חבילות מידע – Wireshark, הרגשתי כי ניתן לעשות זאת בצורה נוחה ואינטואיטיבית יותר בשל עומס ועודף הנתונים אשר ברוב המקרים אינם רלוונטיים לבדיקות שגרתיות של תעבורת הרשת.

הפרויקט הוא מערכת אשר מטרתה לאפשר אבחון וניטור של תעבורת הרשת העוברת דרך המחשב.

כמו כן, מטרת הפרויקט הוא להוות כלי נוח ואינטואיטיבי לאנשי תקשורת על מנת לעבור על תעבורת הרשת בארגון, או אף במערכת הביתית שלהם, ולבודקי חוסן (penetration testers) לחקור את התעבורה במחשב מסוים אשר עליו הוא מבצע בדיקה והפורטים הפתוחים בו, על מנת לאבחן האם קיימות פגיעויות אפשריות הנובעות מהגדרת הרשת במחשב או בארגון.

**תקציר הפרויקט**

**רכיבי המערכת**

1. שרת((Server – השרת מאכסן לי את מסד הנתונים המכיל את נתוני המשתמשים ואת טבלת הבדיקות שכל משתמש עשה עד כה.
2. Controller – מחשב עליו נמצאת התוכנה, וממנו ניתן לבצע פעולות על כל המחשבים הנוספים בLAN המכילים את התוכנה. דרך מחשב זה בלבד מתחברים למשתמש/נרשמים כמשתמש.
3. לקוחות(Clients) – כל המחשבים עליהם מותקנת התוכנה מלבד הController ועליהם מתבצעות הבדיקות בהוראת הController-.

* תקציר כולל ורציונל הפרויקט
* מה המוצר המוגמר אמור לבצע
* למה בחרתי בפרויקט ומה האתגרים שאני צופה לי בפרויקט

1. הגדרת הלקוח

המערכת מכוונת לשני קהלי יעד עיקריים, כאשר שניהם טכנולוגיים בעיסוקם. קהל בודקי חוסן וחוקרי אבטחת מידע , קהל מנהלי הרשתות, אנשי system. הסיבה לכך, היא שהמערכת נועדה לסייע לאנשים בתפקידים אלו לבצע את עבודתם בצורה יעילה ומהירה יותר ובקלות ונוחות רבה יותר, באמצעות הכלים שפיתחתי כחלק מן המערכת.

1. הגדרת יעדים/מטרות

Eagle – Eye מאפשרת את הפעולות הבאות:

* מיפוי העמדות הפעילות ברשת והצגת כתובות הIP שלהן.
* ביצוע בדיקות תעבורה על מספר מחשבים.
* ראיית תוכן חבילות המידע שנקלטו.
* הצגה של המידע בצורה מסודרת על פי פרוטוקולים.
* סינון ומיון הפאקטות על פי זמן הקבלה, כתובת האיי פי ממנה התקבלו, הפורט וכו'..
* סריקת פורטים בכל המחשבים הנשלטים הן פורטי TCP והן פורטי UDP.
* בחינה של תוצאות בדיקות אשר נעשו בעבר דרך המשתמש.
* יצירת Reverse Shell בכל אחד מהמחשבים עליהם מותקנת התוכנה, מה שמאפשר השתלטות מרחוק כל זמן שהעמדה פעילה ומחוברת לController.

1. בעיות, תועלות וחסכונות

הבעיות אשר המערכת שלי פותרת היא סקירה של פגיעות רשת ארגונית מפני תקיפות זדוניות של אנשים המעוניינים בפגיעה בארגון/למען תועלת אישית. כמו כן, היא פותרת בעיה הקיימת בכלים הקיימים לצורך סקירות שכאלו כגון Nmap, Wireshark אשר אינם יעילים, נוחים ואינטואיטיביים עבור סקירה של רשת ארגונית ולא מחשב בודד מה שמקשה על אנשי המקצוע לבחון רשתות במהירות ובמקצועיות. כמו כן, לא מעט צצות בעיות של בלבול ושכחה בין היכולות הרבות והמסובכות/מסורבלות בכלים אלו אשר מעכבים את אנשי המקצוע להשיג את מטרתם במהירות. המערכת מנסה להציע פתרון נוח ופשוט לבעיות אלו באמצעות ממשק משתמש פשוט ואינטואיטיבי שיאפשר ללקוחות לבצע את עבודתם ללא צורך בפקודות מסובכות. המערכת מעניקה מספר שירותים ללקוחותיה, הראשון בהם הוא מסד נתונים המאפשר התחברות של הלקוח לאתר וקבלת תוצאות של סריקות שביצע על רשתות בעבר. כמו כן, המערכת מעניקה כלים להסנפת תעבורת רשת, סריקת פורטים, התחברות לממשק הפקודות של מחשבים וביצוע פעולות טכניות נוספות עליהם, כגון העברת קבצים ושליחת צילום מסך.

1. האם צפויים קשיים או מגבלות בהגדרת המערכת

ישנם מספר קשיים הצפויים במערכת. הראשון שבהם הוא מהירות הסריקה. מכיוון שמדובר במסניף תקשורת אשר נכתב בפייתון שהינה שפת High Level, זמן הפעולה של כל סריקת רשת יהיה איטי באופן יחסי לתוכנית מקבילה בשפת Low Level כגון C או ++C. בנוסף, ניתן יהיה לסרוק רק מחשב אחד בכל פעם מכיוון שאי אפשר להציג את תוצאות הסריקות ביחד באותו דף באתר.

בנוסף, יש צורך בשביל שהפרויקט יעבוד שעל כל המחשבים אותם אנו רוצים לסרוק יפעל השרת שבניתי ויהיה מותקן עליו הפרויקט שלי(לפחות קבצי הbackend).

כמו כן, יש מגבלה נוספת על הפרויקט שהוא עובד אך ורק על מחשבים הנמצאים בLAN, הרשת המקומית. זאת מכיוון שלא יישמתי port forwarding שינתב את כל התקשורת בפורטים בהם משתמש הפרויקט אל מחשב השרת.

* האם מדובר בטכנולוגיה חדשה ובלתי מוכרת?
* האם קיימים סייגים בהגדרת המערכת? (הגבלות שנובעות מבעיות/כשלים/צרכים/ציוד מיוחד... וכו')

1. תיחום הפרויקט

* יש לתאר את התחומים בהם הפרויקט עוסק (בדגש על רשתות ומערכות הפעלה)

הפרויקט עוסק בקשת רחבה של תחומים. הוא עוסק בתחום ה- Web בכך שממשק המשתמש שלי מבוסס על דפי HTML והשרת העיקרי בפרויקט הוא שרת Flask.

כמו כן, הפרויקט עוסק ונוגע בנושאים רבים בתחום אבטחת המידע, בכך שהוא משלב נושאים כגון: Reverse Shell, Encryptions, Hashes, Defense against SQLI, Port Scanning.

בנוסף, הפרויקט עוסק בצורה רחבה מאוד ברשתות. יש לי בפרויקט שלושה שרתים שונים, האחד שרת Flask, שהינו שרת TCP הפועל בפורט 80, הפורט המיועד לתקשורת בפרוטוקול HTTP, בה עובד עולם הWeb. שרת נוסף הינו שרת TCP הפועל בפורט 16549, הקיים על כל מחשבי הרשת, אשר מטרתו לקשר בין הfront end לback end. כל פעולה שנעשית באתר מובילה לפונקציה שנקראת בצד שרת הFlask וכאשר המשתמש מעוניין להפעיל סריקה כזו או אחרת על מחשב ברשת המקומית, ישלח הלקוח הנמצא על המחשב המחובר לאתר הודעה לשרת הנמצא על המחשב עליו המשתמש מעוניין לבצע בדיקה. בנוסף, ישנו גם שרת (מאזין) Reverse Shell, העובד בפרוטוקול TCP על פורט 9999 והינו בעל יכולות רבות: העברת קבצים, העברת צילום מסך, ביצוע פקודות shell והעברת התוצאות. נוסף לכל, הפרויקט מכיל מסניף תקשורת, אשר קולט את חבילות המידע הנשלחות ברשת המקומית ומציג למשתמש את התעבורה בפרוטוקולים הנמצאים בשימוש הרב ביותר והינם בסיכון הגבוה ביותר למתקפות רשת: SSH, ICMP, DNS, HTTP, DHCP, SMB וFTP.

**פרק ב' - 'שם המערכת' – אפיון**

1. פרוט המערכת:



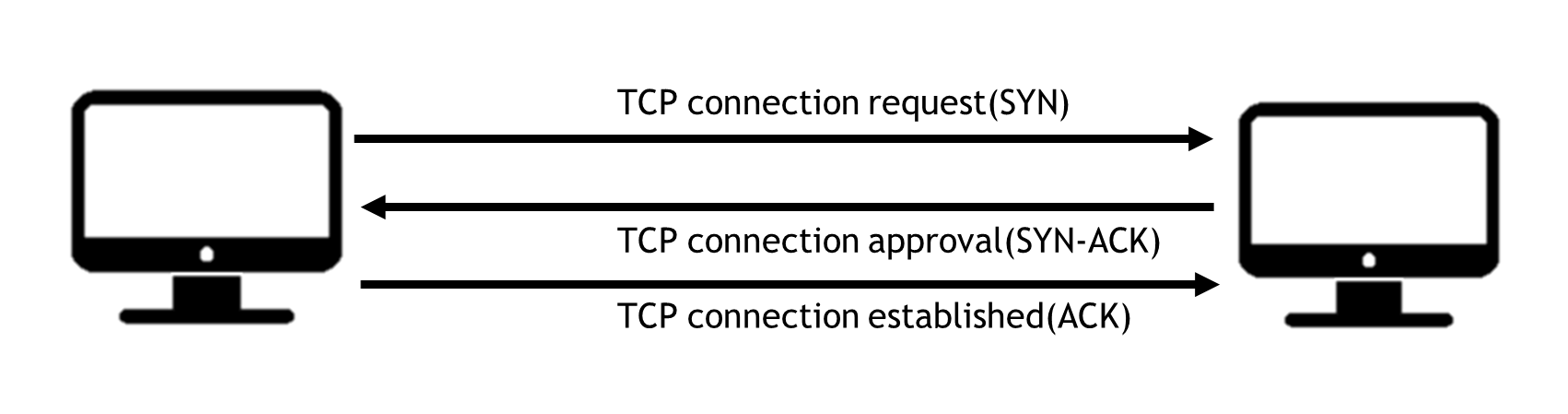
המערכת בנויה בארכיטקטורה של שרת לקוח כאשר ישנם 3 שרתים ולקוחות מסוגים שונים המאפשרים את ביצוע הפעולות על המחשבים ברשת בצורה טובה ויעילה. השרת הראשון הינו שרת הFlask האחראי על שליחת וקבלת בקשות הHTTP למשתמש והצגת דפי הHTML של האתר למשתמש. כמו כן, השרת מתקשר עם מסד הנתונים וקיימים עליו קבצי הסנפה מסוג PCAP עליהם נמצאות תוצאות של הסנפות שבוצעו בעבר על ידי המשתמש. בנוסף, ישנו שרת TCP הכתוב במודול socket ותפקידו הוא לרוץ על כל המחשבים ברשת ולחכות לבקשה להרצת סריקה מצד המשתמש. נוסף לכל, ישנו שרת Reverse Shell אשר מטרתו היא לבקש מאחד המחשבים חיבור ישיר למחשב השרת, ובאמצעות חיבור זה להעביר פקודות shell ופקודות נוספות לביצוע ולקבל את פלטן. פלט הפקודות יעבור באמצעות הAPI אל שרת הFlask אשר יכניס את הפלט אל דפי האתר וישלח למשתמש על גבי פרוטוקול הHTTP.

המערכת אמורה לאפשר לבודקי חוסן במצב בו הם צריכים לבחון את אבטחתם של המחשבים ברשת מקומית כלשהי לעשות זאת במהירות וביעילות, באמצעות הכלים שפיתחתי ושולבו במערכת. בין הכלים נמצאים סורק רשת, המציג למשתמש את העמדות הפעילות ברשת, ומכך הבודק יכול להבין שישנה בעיה בתקשורת/העמדה כבויה ולכן כתובת הIP של אחד המחשבים ברשת אינו נמצא ברשימה. סורק הרשת מבוסס על Pingים שאלו בעצם פאקטות העוברות בפרוטוקול הICMP שמטרתו היא לוודא את קישוריות העמדות ברשת ולבחון שהינן מצליחות לתקשר אחת עם השנייה. בנוסף, ישנם סורקי פורטים במערכת משני סוגים: סריקת SYN שמטרתה היא להיות מהירה ולחסוך במשאבים, וסריקת Stealth שמטרתה להיות חשאית יותר ולא להיחסם על ידי חומות אש. אז כיצד שתי הסריקות מצליחות להשיג את מטרתן?

נתחיל עם הסבר קצר על פרוטוקול הTCP וכיצד נוצר חיבור רציף המאפשר תקשורת בין שני מחשבים.

כאשר מחשב רוצה ליצור תקשורת TCP עם מחשב אחר ראשית הוא שולח הודעת SYN, המעידה על בקשה ליזימת תקשורת. לאחר מכן, אם המחשב השני פנוי ומוכן לביצוע התקשורת, הוא ישלח הודעת SYN ACK, ולבסוף המחשב שיזם את התקשורת ישלח הודעת ACK אשר מאשרת את קבלת האישור ותחילת ההתקשרות.

תהליך זה נקרא TCP 3 way handshake.



סריקת הפורטים מסוג SYN מבוססת על לחיצת היד המשולשת המתבצעת בפרוטוקול TCP וחוסכת בזמן בכך שהיא אינה משלימה את לחיצת היד עד הסוף, אלא שולחת פאקטה עם הדגל SYN המעיד על התחלת לחיצת היד המשולשת ובמידה והתקבלה הודעה מהשרת עם דגל הSYN – ACK הסריקה תסמן את הפורט כפתוח ותמשיך לפורט הבא ללא השלמת החיבור ושליחת הודעת הACK.

סריקת הפורטים מסוג Stealth מבוססת על לחיצת היד המשולשת גם היא, אך בניגוד לסריקה הקודמת שאינה משלימה את לחיצת היד עד הסוף, בכך שאינה שולחת את הודעת הAcknowledge (ACK) זו שולחת הודעה לשרת לאחר קבלת הSYN – ACK עם דגל הנקרא RST(Reset), אשר מעיד על כך שנפל החיבור בין השרת ללקוח, בשל שגיאה פטאלית. בכך חומות אש אשר נועדו להגן כנגד סריקות כאלו, (שיש לציין נחשבות ללא חוקיות במידה ואינן מבוצעות באישור של בעל המערכת/המחשב) לא יחסמו את כתובת הIP של המחשב ממנו נשלחה הבקשה, מפני שיחשבו שמדובר בתקלה בחיבור ולא בתוקף אשר ביצע סריקת פורטים על שרתי הארגון.

בנוסף לסריקות אלו, המערכת מסוגלת לבצע סריקת פורטי UDP העובדת בצורה הבאה. בניגוד לפרוטוקול TCP, פרוטוקול UDP אינו מבוסס חיבור, connection-less, לכן, על מנת לקבוע האם הפורט פתוח, נשלח בקשה לשרת בפורט אותו אנו בודקים, ובמידה והתקבלה תשובה בפרוטוקול UDP מן השרת באותו הפורט ניתן להסיק כי הפורט הינו פתוח.

בנוסף לכל, המערכת מאפשרת הסנפה של התקשורת הנכנסת אל המחשב או הנשלחת בbroadcast, לדוגמה בWi-Fi. רחרחן הרשת מסנן את התקשורת אך ורק לפרוטוקולים הבאים: SMB, FTP, DHCP, ICMP, HTTP, DNS, SSH. הסיבה שבחרתי את כל אחד מהפרוטוקולים הללו להיות אלה שהרחרחן יפלטר היא שאלו הפרוטוקולים הנפוצים ביותר לניצול בעת תקיפה והם מסוכנים מאוד. לכן, במידה וישנה תקשורת חריגה בפרוטוקולים אלה כפי שיראה בודק החוסן, ידע להסיק מכך שככל הנראה ישנה פגיעות באחד הפרוטוקולים הללו או באחד מן השירותים שהשרת מספק. על מנת להמחיש את הסכנה הטמונה בכל אחד מהפרוטוקולים הבאים אציין מספר מתקפות/חולשות הקיימות בפרוטוקולים אלו.

SMB – EternalBlue, חולשה שנותנת הרצת קוד על מחשב בלי להתחבר אליו, נובעת מהדרך שבה מערכת ההפעלה Windows מתנהלת עם פאקטות הSMB, מה שמאפשר לתוקפים אפשריים לשלוח פאקטות זדוניות שיובילו ליכולת הרצת קוד מרחוק על המחשב. יש לציין שהחולשה קיימת על הגרסה הראשונה של הפרוטוקול, SMBv1.

FTP – פרוטוקול הFTP הינו פרוטוקול להעברת קבצים העובד בפורט 21, ומעצם היותו פרוטוקול להעברת קבצים ישנה סכנה רבה בו, שיועברו דרכו קבצים זדוניים. בFTP ישנה אפשרות להתחברות אנונימית מה שיאפשר לתוקף אפשרי ללא סיסמה אל השרת להיכנס לשירות ולהעביר קבצים זדוניים לשרת, אשר יאפשרו לו הרצת קוד על השרת. פעמים רבות אפשרות זו כבויה אך הסיסמה ושם המשתמש שמכניסים לפרוטוקול הינם דיפולטיביים/נפוצים, מה שמאפשר לתוקף להריץ מתקפת מילון על מנת למצוא את שם המשתמש והסיסמה המתאימים. מתקפה שכזו, יוצרת תקשורת רבה ו"רעש" רב ולכן הסנפה של התעבורה בפורט 21 יכולה להועיל במציאת נסיון תקיפה על שירות הFTP.

DHCP - CVE-2019-0547 הינה חולשה שגולתה בשנת 2019 המאפשרת הרצת קוד מרוחק במידה ושולחים פאקטות DHCP זדוניות אל לקוח.

HTTP – פרוטוקול זה הינו הפרוטוקול עליו מבוססים אתרי האינטרנט, אך פרוטוקול זה אינו מוצפן, לכן במידה ויש אתר העובד עם פרוטוקול HTTP ולא HTTPS המוצפן, תוקף פוטנציאלי יוכל להסניף תקשורת זו ולראות את תוכן דפי הHTML המועברים.

ICMP – הפרוטוקול האחראי לבדיקת תקשורת בין עמדות ברשת. משתמש בפאקטות הנשלחות בפרוטוקול זה על מנת לוודא שישנו חיבור תקין בין שתי העמדות. בהסנפת תקשורת זו יש שני יתרונות: האחד, בודקי החוסן וטכנאי הרשת, יוכלו לוודא שהפינגים, פאקטות הICMP אכן מגיעות למחשב. השני הינו שישנה מתקפה ידועה המבוצעת בפרוטוקול זה, ICMP Flood, מתקפה מסוג DOS/DDOS שמטרתה להקריס את שרתי הארגון באמצעות שליחה של פאקטות ICMP רבות.

DNS – פרוטוקול זה אחראי על המרת שמות דומיין לכתובות IP. מה שמאפשר למשתמש הקצה לזכור שמות דומיין ולא כתובות IP ובכך מקל עליו. לדוגמה, שרת הDNS של google יחזיר לראוטר את כתובת הIP של השרת האידיאלי ביותר למענה של גוגל לאחר שאחד מהמחשבים ברשת המקומית ביקש להתחבר למנוע החיפוש של google. בשרת הDNS נמצאים records של השרתים השייכים לארגון, כך שידע השרת להפנות את המשתמשים אל כתובות הIP המתאימות. בכך שרתי DNS יכולים לספק מידע ואינפורמציה טובה לתוקפים על היקף הארגון וכתובות הIP של השרתים, עליהם יוכלו ברגע שידעו את כתובת הIP לבצע סריקות ובחינת וקטורי תקיפה.

SSH – פרוטוקול זה פועל בפורט 22 והינו פרוטוקול חשוב ושימושי מאוד המאפשר התחברות מרחוק לממשק הפקודה של מחשב בתקשורת מוצפנת. הפרוטוקול משתמש באלגוריתמי הצפנה חזקים כגון AES וכן גיבובים כמו SHA-2 על מנת לוודא את שלמות המידע שהועבר. תוקפים פעמים רבות משתמשים בפרוטוקול זה על מנת להתחבר למחשב מרוחק לאחר שמצאו פרטי התחברות בדרך כזו או אחרת או באופן לוקאלי – מקומי או על ידי SSH Tunneling, שיטה בה ניתן להעביר תקשורת SSH ברשת הWAN, בכך שבעצם מפנים את התקשורת הנכנסת אל השרת בפורט מסוים שהוגדר לפורט בו נמצא השירות. או לעתים נדירות יותר, לאחר שמצאו חולשה בפרוטוקול(נכון לגרסאות הישנות של SSH).

בנוסף, משתמשים בSSH Tunneling הרבה כרגל ברשת, או כשרת )socksפרוטוקול אינטרנטי, אשר מעביר חבילות מידע בין שרת ללקוח דרך שרת Proxy).

לדוגמה, אם יש שרת ברשת פנימית שמריץ שרת Web, ולתוקף יש גישה למחשב ברשת, הוא יעדיף שלא להתחבר אליו באמצעות פרוטוקול ה- SSH ולהשתמש בכלים לשליחת בקשות HTTP כמו curl, אלא להשתמש בTunneling עם פורט 80/443 דרך SSH ככה שיוכל לגשת לכל כתובות הIP הפנימיות וככה לגשת לשרת ה - Web בצורה נוחה דרך הדפדפן.

בכך שהמשתמש יסניף את התעבורה באמצעות רחרחן הרשת שפיתחתי כחלק מהמערכת, אשר מסנן בין היתר חבילות מידע העוברות בפרוטוקול הSSH, יוכל להבחין המשתמש האם ישנו מצב חריג של התחברות מרחוק לעמדה בכך שיראה כתובת IP חיצונית המתחברת לשירות, או Tunneling כלשהו על גבי SSH וידע שעליו לנקוט בצעדים להגנה על התחנה.

נוסף לכל, הפיצ'ר האחרון אותו שילבתי במערכת הינו חיבור הReverse Shell. לפני שאצלול אל היכולות והפיצ'רים של הReverse Shell אסביר קודם כל מהו חיבור Reverse Shell, במה הוא שונה מחיבור TCP סטנדרטי, ומהי מטרתו. על מנת להבין מהו Reverse Shell, עלינו להבין קודם כל מהו חיבור Bind Shell.

חיבור Bind Shell הינו חיבור TCP רגיל שבאמצעותו מועברות פקודות Shell לביצוע, ותוצאות הרצתן על המחשב המרוחק. במצב כזה, יש צורך בשתי כתובות IP חיצוניות. בואו נניח שלאליס יש בעיה בפרטי המשתמש שלה והיא צריכה עזרה מבוב. בוב מבקש להתחבר לshell במחשב של אליס באופן ישיר ואליס מאפשרת לו להתחבר. כך, המחשבים יוצרים חיבור באמצעות לחיצת היד המשולשת בפרוטוקול TCP.

כעת, דמיינו שעומדת חומת אש בין הראוטר למחשב של אליס, חומת האש לא תאפשר לחיבור הישיר עם בוב להתבצע. לכן, פותחה שיטת החיבור הנקראת Reverse Shell, אשר מאפשרת מעקף של חומת האש.

בעצם, במקום שבוב יבקש מאליס להתחבר ל shell על המחשב שלה, בוב יפתח שרת(נקרא גם מאזין) על המחשב שלו ואליס תבקש להתחבר לשרת על המחשב של בוב. כך, חומת האש לא תחסום את התעבורה ובוב יוכל להתחבר לממשק הפקודה על המחשב של אליס ולעזור לה לפתור את הבעיה.

תמונה שמכילה טקסט, צילום מסך, אלקטרוניקה, מחשב

התיאור נוצר באופן אוטומטי



הReverse Shell שבניתי בעל יכולות רבות שנועדו להקל ולאפשר פונקציונליות רבה לבודקי החוסן, במידה וירצו להתחבר לאחד מן המחשבים ברשת.

ראשית, ניתן להעלות קובץ מן השרת ללקוח ולהוריד קובץ מהלקוח לשרת. ניתן לערוך חיפוש של קובץ בנתיב מסוים(תיקייה או כונן), חיפוש של כל התיקיות בנתיב מסוים, העברת צילום מסך מהמחשב של הלקוח לשרת, חיפוש כל הקבצים עם סיומת מסוימת בנתיב, וביצוע פקודות shell. התקשורת המועברת באמצעות הReverse Shell מוצפנת באמצעות הצפנה היברידית המבוססת על שני אלגוריתמי ההצפנה החזקים: AES וRSA. RSA הינה הצפנה אסימטרית, כלומר לכל צד ישנו מפתח ציבורי ופרטי, כל צד שולח לצד השני את המפתח הציבורי שלו, ובעת שליחת הודעה הוא מצפין את המידע עם מפתח זה. הצד השני כאשר מקבל את ההודעה מפענח את תוכנה באמצעות המפתח הפרטי שנמצא ברשותו.

מצד שני, הצפנת הAES היא הצפנה סימטרית, מה שאומר שישנו מפתח הצפנה אחד יחיד, אשר משמש גם לצורך הצפנת המידע וגם לצורך פענוחו. הצפנה זו משתמשת במפתח באורך 256 ביטים (32 בתים).

בהצפנה היברידית אנחנו משתמשים גם ב-AES וגם ב RSA-על מנת להעביר מידע בצורה מאובטחת.

בתחילת החיבור ללקוח נשלח לו את המפתח הייחודי שיצרנו בשרת(מפתח הAES) באופן מוצפן עם המפתח הציבורי של הלקוח בהצפנת הRSA. לאחר שיש ללקוח את המפתח הייחודי שיצרנו בשרת נוכל להעביר מידע בצורה תקינה ללא בעיות של גודל טקסט כפי שהיה לנו בהצפנת הRSA. על ידי שימוש בשני סוגי ההצפנות (סימטרי ואסימטרי) אני מתגבר על הקשיים שיש בשימוש בכל אחת מההצפנות בנפרד.

בשיטה הסימטרית הקושי היה באבטחה כי מדובר על אותו מפתח להצפנה ופיענוח ובמידה ולמישהו יש את המפתח, המידע חשוף. בשיטה האסימטרית יש אבטחה מעולה אבל ישנו קושי בלקבל מידע שלם במידה והמידע ארוך (מעל 470 תווים(.

בנוסף לכל, יישמתי פיצ'ר נוסף הקיים בממשק הפקודה בווינדווס ולינוקס, הנקרא history feature. מה שמאפשר למשתמש באמצעות לחיצה על מקשי החצים להסתכל ולהריץ את הפקודות שהורצו קודם לכן. אני עושה זאת בכך שכל פקודה שהשרת שולח נשמרת ברשימה באובייקט. בעצם מופעל Listener של pynput וכאשר נלחץ מקש האנטר נשמרת הפקודה ברשימה והפקודה הנוכחית מתאפסת. כאשר אחד החצים נלחץ, הפקודה האחרונה נשלפת מהרשימה באמצעות Pop, נמחקת הפקודה הנוכחית, ומוקלדת הפקודה הקודמת.

**פונקציונליות המערכת**

1. הרשמה ראשונית למערכת.

2. התחברות משתמש קיים למערכת (Log In).

3. אימות משתמש דרך מייל.

4. אפשרות איפוס סיסמה.

5. אפשרות צפייה בסריקות קודמות.

6. אפשרות לסרוק את העמדות הפעילות ברשת המקומית.

7. אפשרות לבצע סריקת פורטים מסוג SYN TCP על כל אחד ממחשבי הרשת.

8.  אפשרות לבצע סריקת פורטים מסוג Stealth TCP על כל אחד ממחשבי הרשת.

9. אפשרות לבצע סריקת פורטי UDP על כל אחד ממחשבי הרשת.

10. אפשרות הפעלת הסנפת תקשורת על כל אחד ממחשבי הרשת.

11. אפשרות התחברות לממשק הפקודה של כל אחד מהמחשבים באמצעות Reverse Shell.

12. התנתקות מהמערכת( (Log out.

**דרישות פונקציונליות**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **דרישה** | **קלט** | **תהליך** | **פלט** | **טיפול בשגיאות** |
| **1** | הרשמה ראשונית למערכת – רישום של משתמש חדש | נתוני המשתמש החדש (שם פרטי, שם משפחה, אימייל, שם משתמש נבחר, סיסמה נבחרת). | העברת נתוני המשתמש המתקבלים לשרת ושמירתם בבסיס הנתונים( הסיסמה מגובבת). | דף HTML המודיע כי הלקוח נרשם בהצלחה למערכת. | תוחזר הודעת שגיאה במקרה של:  - אימייל לא תקין/תפוס.  - שם משתמש תפוס.  - סיסמה קצרה מידי.  - אימות סיסמה שגוי. |
| **2** | התחברות משתמש קיים למערכת | נתוני המשתמש הקיים (שם משתמש, סיסמה). | העברת נתוני המשתמש המתקבלים לשרת ובדיקתם בבסיס הנתונים. | הודעה המודיעה כי הלקוח התחבר בהצלחה למערכת, ומעבר למסך אימות. | תוחזר הודעת שגיאה אם:  - לפחות מהנתונים שהתקבלו לא מתאים לנתונים הקיימים מראש במסד הנתונים. |
| **3** | איפוס סיסמה | אימייל | ביצוע בדיקת התאמה בין ה ושם המשתמש הנקלטים. אם קיים מייל זה במערכת, ישלח לאימייל קוד אימות לשחזור סיסמה. המשתמש יתבקש להקליד קוד זה, ובמידה והקוד נכון תינתן אפשרות לשינוי סיסמה. | מסך המאפשר איפוס ושינוי סיסמה. | תוחזר הודעת שגיאה במקרה וכתובת האימייל שהתקבלה לא תקינה.  תוחזר הודעה מתאימה במקרה שלא קיים מייל כזה במערכת. |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **דרישה** | **קלט** | **תהליך** | **פלט** | **טיפול בשגיאות** |
| **4.1** | הפעלת הסנפה | מספר הפאקטות להסנפה(ברירת מחדל 1000). | הסנפת תקשורת | הפאקטות שהוסנפו ושייכות לפרוטוקולים הבאים: SSH, ICMP, DNS, HTTP, DHCP, SMB וFTP | אין |
| **4.2** | הפעלת סריקת פורטים SYN | טווח הפורטים לבדיקה | סריקת פורטים | הפורטים הפתוחים | אין |
| **4.3** | הפעלת סריקת פורטים Stealth | טווח הפורטים לבדיקה | סריקת פורטים | הפורטים הפתוחים | אין |
| **4.4** | הפעלת סריקת פורטים UDP | טווח הפורטים לבדיקה | סריקת פורטים | הפורטים הפתוחים | אין |
| **4.5** | התחברות בReverse Shell | פקודה לביצוע | הרצת הפקודה על המחשב המרוחק | תוצאות הפקודה | פקודה לא תקינה, שגיאות חיבור. |
| **5** | התנתקות מהמערכת | בקשת התנתקות מהמערכת של המשתמש. | ניתוק המשתמש מהמערכת. | סגירת הSession של המשתמש והעברה לדף HTML האומר כי התנתק בהצלחה. | אין |

1. מה היכולות שהיא תעניק למשתמש , פירוט היכולות   
   במידה ויש משתמשים שונים, לפרט לפי כל משתמש:

* אפשרות צפייה בסריקות קודמות.
* אפשרות לסרוק את העמדות הפעילות ברשת המקומית.
* אפשרות לבצע סריקת פורטים מסוג SYN TCP על כל אחד ממחשבי הרשת.
* אפשרות לבצע סריקת פורטים מסוג Stealth TCP על כל אחד ממחשבי הרשת.
* אפשרות לבצע סריקת פורטי UDP על כל אחד ממחשבי הרשת.
* אפשרות הפעלת הסנפת תקשורת על כל אחד ממחשבי הרשת.
* אפשרות התחברות לממשק הפקודה של כל אחד מהמחשבים באמצעות Reverse Shell.

1. פירוט הבדיקות ('קופסא שחורה')

(בדיקות לביצוע אך ורק עם קובץ ההרצה, ללא הקוד, בדגש על כך שכל אחד לפי הכתוב בטבלה יצליח לבצע את הבדיקה)

|  |  |  |  |
| --- | --- | --- | --- |
| **מספר** | **שם הבדיקה (שם שיעיד על תוכן הבדיקה)** | **מה אמורה לבדוק** | **איך מתכננים לבדוק**  **(לתאר בפירוט את שלבי הבדיקה)** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. תכנון לוח זמנים לפרויקט

* תכנון לוח זמנים ראשוני
* עבור כל פעילות בלוח: זמן התחלה מתוכנן, זמן סיום מתוכנן , זמן התחלה בפועל, זמן סיום בפועל והערות. יש לכלול את התאריכים שניתנו לכם עבור כל פרק כ'זמן סיום מתוכנן' ואת החודש של הקידוד לפרק לאבנים גדולות ולתת להן זמנים

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **פעילות** | **זמן התחלה מתוכנן** | **זמן סיום מתוכנן** | **זמן התחלה בפועל** | **זמן סיום בפועל** | **הערות** |
| יזום |  |  |  |  |  |
| אפיון |  |  |  |  |  |
| ניתוח |  |  |  |  |  |
| עיצוב |  |  |  |  |  |
| קוד (לפרוט לאבנים גדולות) |  |  |  |  |  |
| גרסה ראשונית |  |  |  |  |  |
| מסמך בדיקות |  |  |  |  |  |
| מדריך למשתמש |  |  |  |  |  |
| גרסה סופית |  |  |  |  |  |
| סגירת תיק פרויקט |  |  |  |  |  |
| מצגת הסבר |  |  |  |  |  |

* **לוח הזמנים – הינו מסמך חי ויש לעדכנו לאורך כל הפרויקט**

1. ניהול סיכונים בפרויקט

* זיהוי סיכוני הפרויקט, ניתוחם ומענה עליהם, הטבלה תמוין מסיכון גבוה לנמוך (הגבוהים באדום, בינוני –כתום , קל – צהוב) סיכון שיוסר יצבע בירוק ויעבור לתחתית המסמך

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **הסיכון** | **פירוט הסיכון** | **רמת הסיכון**  **(קל/בינוני/קשה)** | **תיאור דרכים (לפחות 2 ) להתמודדות עם הסיכון ולהקטין אותו** | **מה בוצע בפועל** | **תאריך** |
| אי עמידה בזמנים | פרויקט לא יושלם | קשה | * הקדמת לו"זים משימות * ארגון הזמן בצורה יעילה * להתחיל בדברים היותר קשים |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* **ניהול סיכונים – הינו מסמך חי ויש לעדכנו לאורך כל הפרויקט**

**פרק ג' - 'שם המערכת' - מסמך ניתוח**

פרוט יכולות המערכת

**יכולות בצד שרת: רנדור דפי HTML ושליחתם ללקוח(שרת הFlask- ), שליחת פקודות לכל אחד מהמחשבים לביצוע אחת מפעולות הבאות: סריקת פורטים, הפעלת לקוח לReverse Shell, הפעלת הסנפה, שמירת תוצאות ההסנפה בקובץ PCAP, יצירת קשר עם מסד הנתונים לצורך שינוי ואחזור נתונים.**

**צד שרת((Flask:**

שם היכולת: הרשמה למערכת

מהות היכולת: רישום משתמש חדש במערכת

אוסף יכולות:

* קבלת נתונים מהלקוח
* בדיקה מול בסיס הנתונים
* גיבוב סיסמא
* הוספה לבסיס נתונים
* החזרת דף תשובה

שם היכולת: התחברות למערכת

מהות היכולת: התחברות משתמש למערכת

אוסף יכולות:

* קבלת נתונים מהלקוח
* בדיקה מול בסיס הנתונים
* החזרת דף תשובה

שם היכולת: איפוס סיסמה

מהות היכולת: איפוס סיסמה של משתמש לפי מייל

אוסף יכולות:

* קבלת נתונים מהלקוח - מייל
* בדיקה מול בסיס הנתונים
* שליחת מייל עם קוד אימות.
* החזרת דף לאיפוס
* קבלת נתוני סיסמה חדשה
* בדיקת האימות.
* שינוי הסיסמה במסד הנתונים.

שם היכולת: אפשרות צפייה בסריקות קודמות

מהות היכולת: הצגת דף עם תוצאות של סריקות שבוצעו בעבר.

אוסף יכולות:

* הצגת סריקות שבוצעו בעבר.
* הצגת תוצאות הסריקה שנבחרה על ידי טעינה מקובץ PCAP.

**צד שרת(שרת 2) :**

שם היכולת: התחלת סריקת פורטי TCP

מהות היכולת: השרת יקבל הודעה מלקוח להתחיל לבצע סריקת פורטי TCP מסוג מסוים על המחשב עליו הוא נמצא ויפעיל את הפעולה המתאימה.

אוסף יכולות:

* קבלת הודעה מלקוח ובהתאם הפעלת סריקת פורטים, החזרת המידע לשרת הFlask.

שם היכולת: התחלת סריקת פורטי UDP

מהות היכולת: השרת יקבל הודעה מלקוח להתחיל לבצע סריקת פורטי UDP על המחשב עליו הוא נמצא ויפעיל את הפעולה המתאימה.

אוסף יכולות:

* קבלת הודעה מלקוח ובהתאם הפעלת סריקת פורטים, החזרת המידע לשרת הFlask.

שם היכולת: התחלת הסנפה

מהות היכולת: השרת יקבל הודעה מלקוח להתחיל לבצע הסנפה של התקשורת שתפעל על המחשב עליו הוא נמצא, יתחיל בהסנפה ויחזיר את תוצאותיה לשרת הFlask אשר יציג אותן על גבי האתר.

אוסף יכולות:

* קבלת הודעה מלקוח ובהתאם התחלת הסנפה.
* החזרת מחרוזת המכילה את המידע הרלוונטי מהפאקטות שהוסנפו ושייכות לפרוטוקולים הבאים: SMB, FTP, DHCP, SSH, DNS, HTTP, ICMP לשרת הFlask.

שם היכולת: הפעלת לקוח Reverse Shell.

מהות היכולת: השרת יקבל הודעה מלקוח להדליק את לקוח הReverse Shell על המחשב עליו הוא נמצא ויפעיל את הפעולה המתאימה.

אוסף יכולות:

* הדלקת לקוח הReverse Shell-.

**צד שרת(שרת 3) :**

שם היכולת: התחברות ב- Reverse Shell למחשב ברשת המקומית.

מהות היכולת: השרת יבקש להתחבר לאחד מהמחשבים ברשת(אחד מהלקוחות) וישלח ללקוח הודעה בהתאם לפקודה שהכניס המשתמש באתר ויקבל את תוצאות הפקודה מן הלקוח ויעבירן לשרת הFlask אשר ירנדר את המידע שהתקבל אל תוך דפי הHTML ויציג אותו למשתמש או יעביר את הקובץ המבוקש ללקוח/ יקבל קובץ מן הלקוח.

אוסף יכולות:

* שליחת הודעה ללקוח עם פקודה להפעלה על מחשב הלקוח.
* קבלת תוצאות הפקודה/ קבלת הקובץ/ שליחת הקובץ.

רשימת אובייקטים: הצפנה/פענוח, תקשורת, בסיס נתונים

**יכולות בצד לקוח:**

(להתחיל לבנות לפי הרשימה של היכולות שציינתם באפיון, שימו לב שכאן תצטרכו לפרט עד הרמה הבסיסית הנדרשת למימוש כל יכולת , שימו לב מדברים ברמת על לא בקוד) :

**צד לקוח - אתר:**

שם היכולת: הרשמה למערכת

מהות היכולת: רישום משתמש חדש במערכת (קליטת פרטיים אישיים נדרשים)

אוסף יכולות:

* ממשק משתמש – מסך הרשמה
* קליטת נתונים
* בדיקת תקינות
* שליחה לשרת בבקשת POST
* קבלת תשובה מהשרת
* הצגת דף תשובה למשתמש

שם היכולת: התחברות למערכת

מהות היכולת: התחברות למערכת (קליטת פרטיים אישיים נדרשים)

אוסף יכולות:

* ממשק משתמש – מסך הרשמה
* קליטת נתונים
* בדיקת תקינות
* שליחה לשרת בבקשת POST
* קבלת תשובה מהשרת
* הצגת דף תשובה למשתמש

שם היכולת: איפוס סיסמה

מהות היכולת: החלפת הסיסמה של המשתמש(קלט נדרש – מייל)

אוסף יכולות:

* ממשק משתמש – מסך להזנת מייל
* קליטת נתונים
* בדיקת מייל מול השרת
* קבלת מייל אימות.
* הכנסת קוד האימות לדף.
* הכנסת סיסמה חדשה.
* בדיקה של תקינות הסיסמה
* שליחה לשרת בבקשת POST
* קבלת תשובה מהשרת
* הצגת דף תשובה למשתמש

**צד לקוח - TCP:**

שם היכולת: שליחת בקשה לסריקת פורטי TCP

מהות היכולת: לחיצה על כפתור באתר שיבקש ממחשב מסוים להתחיל בסריקת פורטים.

אוסף יכולות:

* שליחת הבקשה לשרת.

שם היכולת: שליחת בקשה לסריקת פורטי UDP

מהות היכולת: לחיצה על כפתור באתר שיבקש ממחשב מסוים להתחיל בסריקת פורטים.

אוסף יכולות:

* שליחת הבקשה לשרת.

**צד לקוח 3- TCP:**

שם היכולת: קבלת פקודה משרת והרצתה על המחשב.

מהות היכולת: הלקוח מקבל פקודה בתקשורת מוצפנת מן השרת, מפענח אותה ומריץ אותה על ממשק הפקודה של המחשב, שומר את הפלט של הפקודה, מצפין ושולח חזרה לשרת.

אוסף יכולות:

* קבלת הבקשה מהשרת.
* פענוח הבקשה.
* הרצת הפקודה.
* הצפנת תוצאות הפקודה
* שליחה לשרת.

שם היכולת: שליחת בקשה לסריקת פורטי UDP

מהות היכולת: לחיצה על כפתור באתר שיבקש ממחשב מסוים להתחיל בסריקת פורטים.

אוסף יכולות:

* שליחת הבקשה לשרת.

שם היכולת: שליחת בקשה לסריקת פורטי UDP

מהות היכולת: לחיצה על כפתור באתר שיבקש ממחשב מסוים להתחיל בסריקת פורטים.

אוסף יכולות:

* שליחת הבקשה לשרת.

רשימת אובייקטים: ממשק משתמש, הצפנה/פיענוח, תקשורת, תהליכונים.

**פרק ד' - 'שם המערכת' - העיצוב**

1. תיאור הארכיטקטורה של המערכת המוצעת

החומרה: מחשבים.

ישנם שלושה רכיבים עיקריים בארכיטקטורת הפרויקט: שרת, לקוח ומסד נתונים.



רכיבי התקשורת מחוברים ומקושרים באופן הבא: ישנו מחשב עליו פועל שרת פלאסק הפועל בפורט 80 ולקוח הפועל בפורט 16549. שרת הפלאסק מציג ממשק משתמש למשתמש באמצעות דפי HTML והוא אחראי לניתוב ולניהול הAPI, במידה והמשתמש ירצה להפעיל סריקה/הסנפה על אחד ממחשבי הרשת עליהם מותקנת המערכת, יגרום שרת הפלאסק להפעלת הלקוח ושליחה של בקשה לשרת הרלוונטי על המחשב המרוחק. התקשורת של לקוח זה מוצגת באדום בסרטוט לעיל ותקשורת שרת הHTTP מוצגת בצבע תכלת. במקרה בו המשתמש ירצה להפעיל Reverse Shell על אחד מהמחשבים הדבר יקרה באופן הבא: האתר יציג כפתור שיאפשר למשתמש לבצע פעולה זו ויורה לשרת הפלאסק לשלוח באמצעות הלקוח הנמצא על המחשב בקשה להפעלת הReverse Shell על השרת המרוחק. כאשר יקבל השרת את הבקשה הוא יפעיל לקוח Reverse Shell, אשר יתחבר אל השרת. כעת, כל פקודה שיכניס המשתמש באתר תועבר לשרת הFlask, אשר יעביר את הפקודה לשליחה באמצעות שרת הReverse Shell. כאשר הפקודה תבוצע בצד הלקוח ישלח חזרה הפלט אל השרת ושרת הFlask יציג פלט זה על גבי האתר. התקשורת בין המאזין ללקוח Reverse Shell מופיע בירוק בסרטוט.

1. תיאור הטכנולוגיה הרלוונטית

* פירוט ההחלטות שנלקחו בחשבון בעת בחירת הטכנולוגיה

לקראת בחירת הטכנולוגיה לפרויקט נלקחו בחשבון מספר פרמטרים חשובים אשר משפיעים על איכות הפרויקט, מהותו והייחודיות שלו. מכיוון שמדובר בפרויקט המשלב לא מעט נושאים הקשורים באבטחת מידע חשבתי תחילה לפתח את הפרויקט על מכונת Linux, בשל הנוחות של כלים הקיימים בלינוקס לשם השוואה עם הפעולות שכתבתי. לדוגמה, קיימים לא מעט כלים לסריקת פורטים והסנפת תקשורת בלינוקס כמו גם מדריכים לכתיבת כלים שכאלו על מערכות הפעלה מבוססות יוניקס כמו לינוקס. כמו כן, כאשר עשיתי מחקר על המודול Flask שנועד על מנת ליצור שרת API בצורה קלה ונוחה ראיתי מספר רב של מדריכים המבוססים על מערכות הפעלה מבוססות Unix. בסופו של דבר, החלטתי להשתמש בWindows מכיוון שהמעבדה בבית ספר מבוססת Windows ומבחינת יעילות, במידה ואפתח את הפרויקט אך ורק על מערכת לינוקס, אצטרך להשתמש במכונה וירטואלית על כל אחד מהמחשבים במעבדה מה שיאט משמעותית את זמני הטעינה והסריקה בשל מגבלת המשאבים במכונה וירטואלית. בסופו של דבר, פיתחתי את הפרויקט בצורה גנרית מספיק כך שאינו תלוי במערכת ההפעלה עליה הפרויקט מופעל, אלא ישנו צורך רק שיהיה מותקןPython על המערכת והפעלתי ובדקתי את הפרויקט בעיקר בביתי, על מכונה וירטואלית של Kali Linux המבוססת על Debian.

בחרתי לתכנת את הפרויקט בPython ממספר סיבות. ראשית, מכיוון שהפרויקט משלב תקשורת ברובו המוחלט,פייתון הייתה בחירה טבעית בשל הנוחות שמאפשרת השפה לתכנת סוקטים בשל האובייקטים הגמישים בניגוד לתכנות סוקטים בשפות אחרות כגון C#. כמו כן, אני בקיא ביותר בשפה זו ולכן העדפתי לתכנת פרויקט בהיקף כזה בשפה שאני מכיר בצורה הטובה ביותר. נוסף לכל, פייתון היא שפה הנמצאת בשימוש רב בתעשייה בעיקר בתחומים בהם עוסק הפרויקט: רשתות ואבטחת מידע, היא מתעדכנת כל הזמן עם ספריות ומדריכים חדשים והתיעוד על הספריות מאוד רחב ומקיף ברוב המקרים. תחומי העניין שלי הם אבטחת מידע ורשתות בעיקר, אני מאוד אוהב ומתחבר לנושאים אלו והם מסקרנים אותי מאוד ולכן בפרויקט בחרתי לחקור ולשלב בעיקר קונספטים ופיצ'רים הקשורים באבטחת מידע ורשתות. שפת התכנות Python מתאימה מאוד לתכנות בנושאים אלו משום שהיא מאוד אינטואיטיבית, ורסטילית ובעלת היכולות הדרושות על מנת ליצור את הכלים הללו.   
(שפת תכנות, OS, תקשורת, תחומי עיניין, ועוד...)

1. תיאור מודולים בהם נעשה שימוש 

* סקירת כל המודולים המרכיבים את המערכת וקשרי הגומלין ביניהם:
  1. מודולים שאתם מייבאים – מספיקה שורה אחת המסבירה למה מיועד המודול.
  2. מודולים שלכם - יש לפרט ברמת אובייקטים (שם האובייקט והתכונות שבו) וכן כלל הפעולות שבמודל (כותרות וטענת כניסה ויציאה לכל פעולה – באנגלית כפי שיקראו בתוכנה)

בפרויקט אני משתמש במספר מודולים שונים בפייתון אשר מאפשרים לי לבצע פעולות מסוימות.

ניתן לחלק אותם לארבע קבוצות של ספריות: ספריות תקשורת, ספריות הצפנה וגיבוב, ספריות מערכת וספריות כלליות.

ספריות תקשורת:

1. Scapy – מודול אשר נועד לבניית, הסנפת ושליחת פאקטות.
2. Socket – מודול אשר מאפשר יצירת שרתים ולקוחות וליצור חיבור סוקט ביניהם.
3. Smtplib – מודול המאפשר שליחה של מיילים מכתובת אחת לכתובת אחרת.
4. Flask – מודול המאפשר יצירת שרת HTTP וניתובים בAPI.

ספריות מערכת:

1. os – מודול המאפשר גישה לפעולות מערכת.
2. threading – מודול המאפשר להריץ פונקציה בפייתון כת'רד.
3. subprocess – מודול המאפשר הרצה של פקודות shell במערכות לינוקס ווינדווס.

ספריות כלליות:

1. time – הספרייה מאפשרת לנו להשיג מידע על הזמן הנוכחי בנקודות מסוימות בקוד.
2. String – הספרייה מאפשרת לנו להשתמש בנוחות ברשימות של כל התוים האפשריים בחלוקה לקטגוריות.
3. Random – הספרייה מאפשרת פונקציות הבוחרות באופן רנדומלי אלמנטים או מגרילות מספרים.
4. PIL – הספרייה מאפשרת פתיחה, עריכה ושמירה של תמונות באמצעות פייתון.

ספריות הצפנה וגיבוב:

1. Cryptodome – הספרייה מאפשרת שילוב של הצפנות של מידע.
2. Hashlib – הספרייה מאפשרת שילוב של גיבוב מידע.

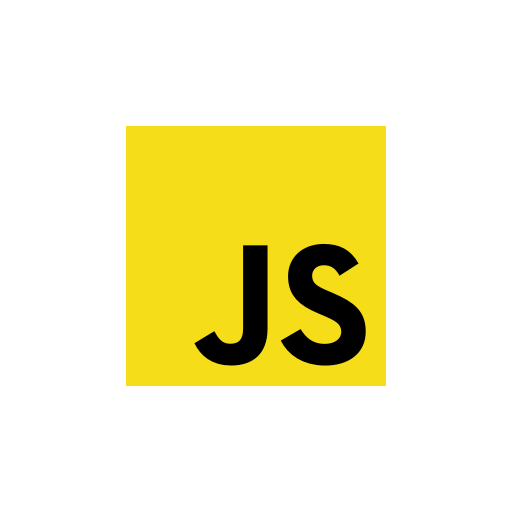
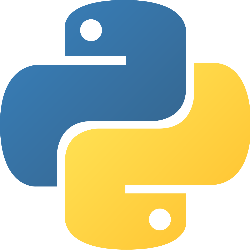
המודולים שאני פיתחתי:

1. Webshell\_Server -> Server object במודול זה יצרתי את מחלקת השרת המתחבר בReverse Shell ללקוח.
2. Helper\_methods -> helpful functions such as take screenshot, iterate over a subnet mask, get IP address, scan a network, Encrypt and Decrypt messages in both client and server side.
3. PortScanner -> PortScanner object – אובייקט אשר מבצע סריקת פורטים.
4. PACKET\_SNIFFER -> gen\_sniff function – הפונקציה מסניפה 1000 פאקטות ומחזירה את הפאקטות ממויינות ברשימה כך(משמאל לימין):

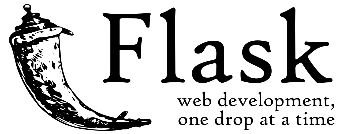
*HTTP, ICMP, SMB, FTP, SSH, DNS, DHCP*

1. Server -> Server object - במודול זה יצרתי את מחלקת השרת הפועל על כל המחשבים ברשת המקומית.
2. Client -> Client object – במודול זה יצרתי את מחלקת הלקוח הפועל על מחשב השרת המרכזי – שרת הפלאסק.
3. תיאור סביבת הפיתוח

* השפה בה תכנתתי את המערכת היא Python.
* כלי הפיתוח הנדרשים הם סורק פורטים, מסניף תקשורת וReverse Shell.
* סביבת הפיתוח שלי היא Pycharm לbackend שנכתב ב Python וVScode לפיתוח הfront end בHTML וJavascript.







1. תיאור האלגוריתמים המרכזיים בפרויקט: (לבחור 4 בעיות עיקריות בפרויקט ולנתח לפי האמור בסעיף זה)

הבעיה האלגוריתמית המורכבת ביותר שנתקלתי בה בפרויקט הינה חלוקה של מספר הפורטים לסריקה לכמה טווחים קטנים יותר לסריקה כמספר ליבות המחשב כפול 2.

בסופו של דבר, החלטתי ליצור רשימה אחת שמספר האיברים בה יהיה מספר הליבות כפול 2 ובכל איבר יהיה טאפל עם הפורט הראשון לסרוק והאחרון לסרוק.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

1. תיאור מסכי הפרויקט:

* לכל מסך מיועד בפרויקט (שרטוט ידני, בהמשך יצורפו מסכי המערכת):
  1. מה מטרת המסך
  2. שרטוט של המסך (מה כולל)
  3. ולהיכן הוא מוביל (ברמת מסכים)

1. תיאור פרוטוקול התקשורת

כל הודעה בתקשורת נשלחת בצורה הבאה, ראשית נשלח אורך המידע שעתיד ישלח ולאחר מכן נשלח המידע.

לקוח -> שרת (Client -> Server)

SRT\_SNF

תיאור: פקודה שמבקשת להתחיל הסנפה של חבילות מידע ברשימת הכתובות.

פקודה: SNF\_SRT

דוגמה להודעה אמיתית בפרוטוקול:

SNF\_SRT

REV\_ACT

תיאור: הפעלת הסוס הטרויאני על מנת ליצור רברס של.

פקודה: REV\_ACT

דוגמה להודעה אמיתית בפרוטוקול:

REV\_ACT

EXIT

תיאור: כיבוי הסוס הטרויאני על מנת להפסיק את החיבור.

פקודה: EXIT

דוגמה להודעה אמיתית:

EXIT

SYN\_SRT

תיאור: הפעלת סריקת הפורטים על פורטי הTCP.

פקודה: SYN\_SRT

דוגמה להודעה אמיתית:

SYN\_SRT

STEALTH\_SRT

תיאור: הפעלת סריקת הפורטים על פורטי הTCP.

פקודה: STEALTH\_SRT

דוגמה להודעה אמיתית:

STEALTH\_SRT

UDP\_SRT

תיאור: הפעלת סריקת הפורטים על פורטי ה-UDP.

פקודה: UDP\_SRT

דוגמה להודעה אמיתית:

UDP\_SRT

1. תיאור מבני הנתונים

יש לי מסד נתונים אחד ותיקייה נוספת עם קבצי PCAP, אשר בה שמורות תוצאות ההסנפות. שניהם קיימים באופן מקומי על השרת ברשת ומאפשרים למשתמש לקבל מהם מידע. מהקבצים – סריקות קודמות ומהמסד נתונים לצורך הרשמה והתחברות.

שדות עבור מסד הנתונים: שם פרטי, שם משפחה, שם משתמש, מייל וסיסמה.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Email | Password | Username  Primary key | Last Name | First Name | Table name: profile |
| VARCHAR  ofekerez@gmail.com | VARCHAR  Fdgsfguiegu123235i | VARCHAR  Ofek123123 | VARCHAR  erez | VARCHAR  ofek | DB name: site.db |

1. סקירת חולשות והאיומים   
   למערכת ואת הפתרונות שלכם לאיומים, יש לסקור ע"פ מאפייני המערכת השונים:

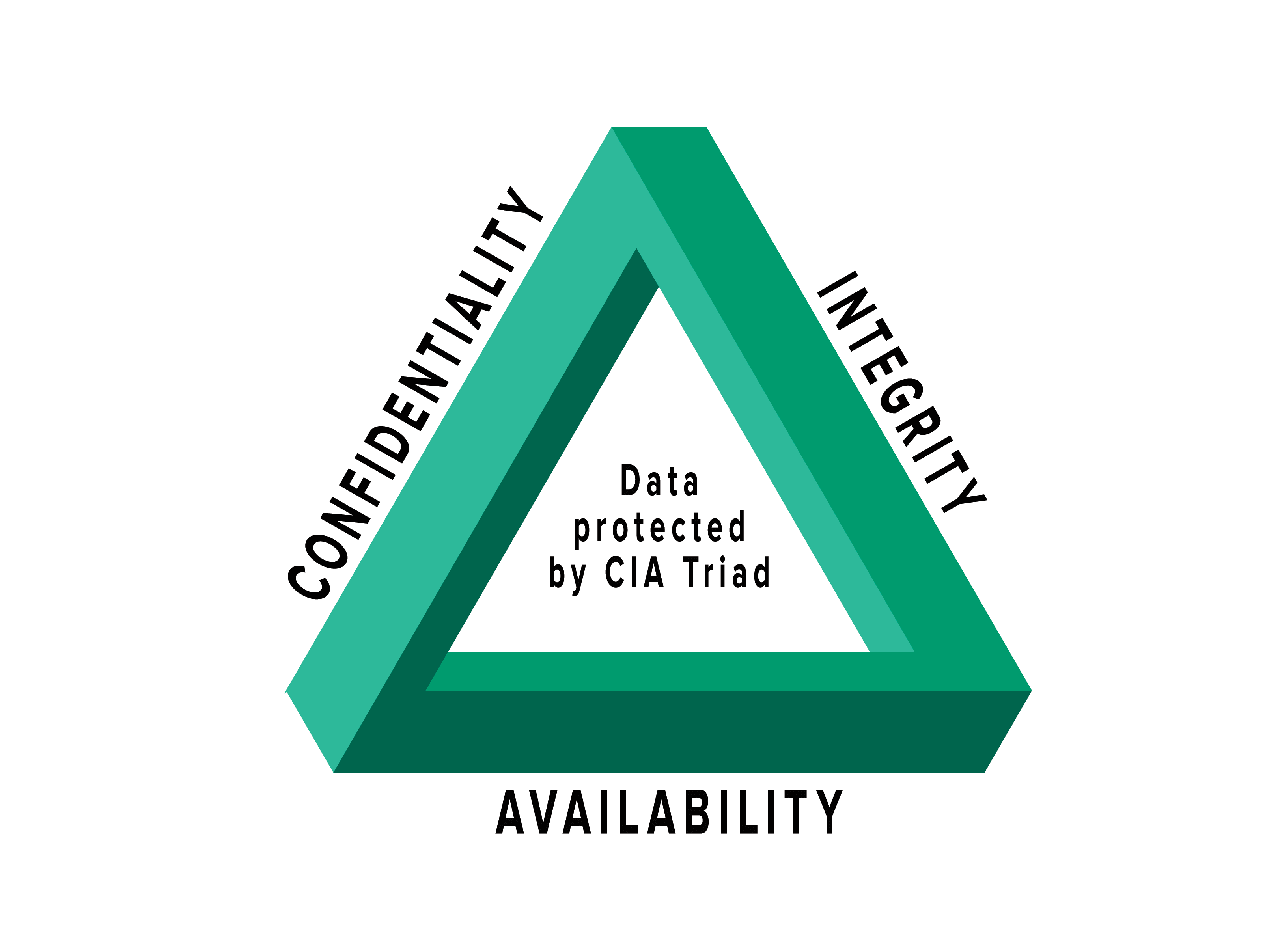
שכבת האפליקציה:

* עבודה עם בסיס נתונים sql injection ...
* תהליך ה login - ישנו אימות על ידי שליחת מייל ובדיקה בצד שרת של תקינות הקוד.
* MITM : מכיוון שהשרת עובד בפרוטוקול HTTP ולא HTTPS התקשורת אינה מוצפנת ותלויה בcertificate שנוצר מפרוטוקול הרשת SSL/TLS ומתקפה זו הינה אפשרית ותציג לתוקף את תוכן דפי הHTML .
* DOS/DDOS : יכולה להיות מתקפת DOS או DDOS מכיוון שבמידה ומשתמש או מספר משתמשים ברשת המקומית ישלחו בקשות רבות לשרת הוא לא יחסום את המחשבים הללו ולא יגביל את כמות הבקשות שהם יכולים לבקש. לכן סביר ביותר, שהשרת לאחר זמן מה, יקרוס.

שכבת התעבורה:

ישנה הצפנה היברידית RSA + AES בReverse Shell על מנת לשמור על סודיות המידע הרגיש העובר בתקשורת המכיל תוצאות של פקודות מערכת.

האתר חסין בפני SQL Injection מפני שאני עובד במסד נתונים עם המודול Flask-SQLAlchemy אשר יוצר באופן אוטומטי שאילתות מאובטחות ועל מנת להכניס או למשוך מידע מהבסיס נתונים יש צורך רק להשתמש בפעולות המובנות במודול.



**פרק ה' - 'שם המערכת' - הקוד**

1. עבור (שעושים קטעי קוד מיוחדים משהו מיוחד, משהו מסובך, משהו בדרך שונה, משהו יפה בעיניכם ויש לכלול בתוכם את קטעי הקוד הרלוונטים לאלגוריתמים המרכזיים שציינתם בפרק הקודם):
2. פונקציה אשר מחלקת את טווח הפורטים לסריקה לרשימה באורך כפול ממספר ליבות המעבד כשכל אלמנט הוא טאפל שמסמל פורט ראשון לסריקה ופורט אחרון לסריקה.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

1. פונקציה אשר מסניפה את התקשורת, שומרת בקובץ PCAP, מפלטרת אותן לפי פרוטוקולים ומכניסה לרשימה ממויינת לפי הפרוטוקולים.

תמונה שמכילה טקסט, צג, צילום מסך, אלקטרוניקה

התיאור נוצר באופן אוטומטי

1. סריקת פורטיTCP בסריקת SYN עם ת'רדים.

תמונה שמכילה טקסט, צילום מסך, צג

התיאור נוצר באופן אוטומטי

1. סריקת פורטי TCP בסריקת Syn/Stealth עם ת'רדים.

תמונה שמכילה טקסט, צילום מסך, צג

התיאור נוצר באופן אוטומטי

1. סריקת פורטי UDP עם ת'רדים.

תמונה שמכילה טקסט, צילום מסך, צג, מסך

התיאור נוצר באופן אוטומטי

1. הרצת פקודת מערכת דרך חיבור reverse shell.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

1. אימות סיסמה באתר.

תמונה שמכילה טקסט, צילום מסך, מחשב

התיאור נוצר באופן אוטומטי

קישור לשאר הקוד של המערכת:

<https://github.com/ofekerez/Eagle-Eye-Project.git>

**פרק ו' - 'שם המערכת' – בדיקות ('קופסא לבנה')**

1. פירוט הבדיקות

* פירוט הבדיקות שהופיעו בפרק האפיון ברמת הביטים – לתאר בפירוט מה נדרש לבצע ומה בוצע בפועל. וכמובן על בדיקה צריכה להתבצע יותר מפעם אחת ועד שהיא עוברת

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **שם הבדיקה** | **מטרת הבדיקה** | **מה נדרש לבצע** | **מתי** | **מה בוצע בפועל** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* יש לתעד כל בדיקה שמבוצעת, חייבים לחזור לפחות פעמיים על כל בדיקה – כל חזרה תתועד בשורה נפרדת עם תוצאות הבדיקה

**פרק ז' - 'שם המערכת' – מדריך למשתמש**

1. מדריך התקנה (סביבת עבודה)

על מנת להתקין את המערכת יש ללכת אל הrepository של הפרויקט הנמצא בכתובת הבאה: <https://github.com/ofekerez/Eagle-Eye-Project.git>

לאחר מכן, יש ללחוץ על Code ולהעתיק את הURL ללוח. אחר כך, יש לפתוח את הCMD או את הגיט bash ולהקליד git clone {URL}, פעולה זו תוריד את קבצי הrepository לתיקייה בה ממשק הפקודה שפתחתם נמצא.

לאחר מכן יש להריץ את הקובץ main.py בתיקיית API, מה שיפעיל את שרת הFlask.

לאחר מכן, עלייך לעבור על כל מחשב ברשת המקומית ולהוריד עליו גם כן את המערכת מהגיטהאב. לאחר מכן עליך להפעיל בו את הקובץ Server.py הנמצא בתיקיית bin.

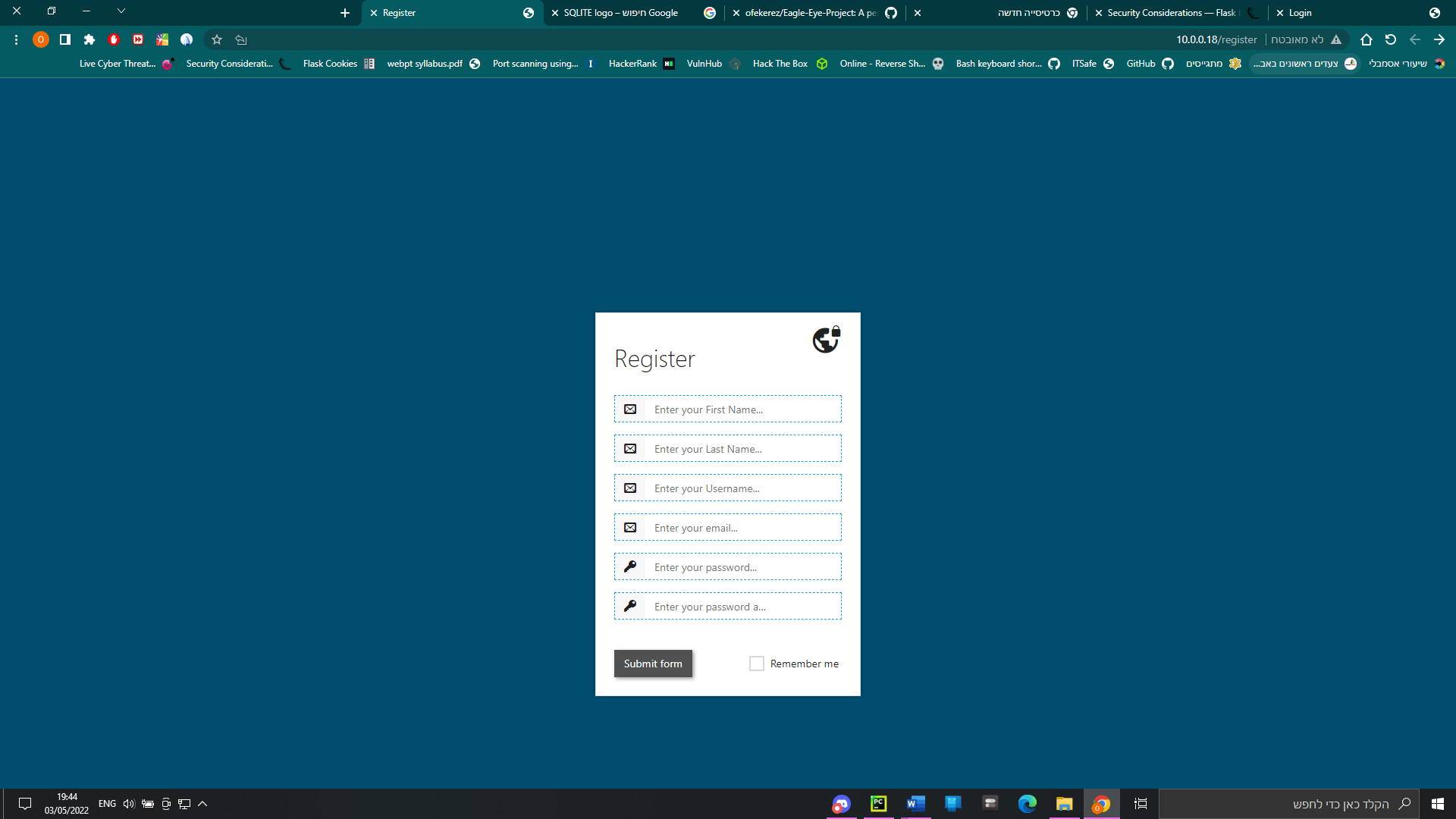
לאחר מכן, תוכל להתחבר ולהפעיל את האתר בצורה מלאה מכל מחשב אחר ברשת.

1. מדריך למשתמש הכולל עבור כל תהליך/יכולת במערכת:

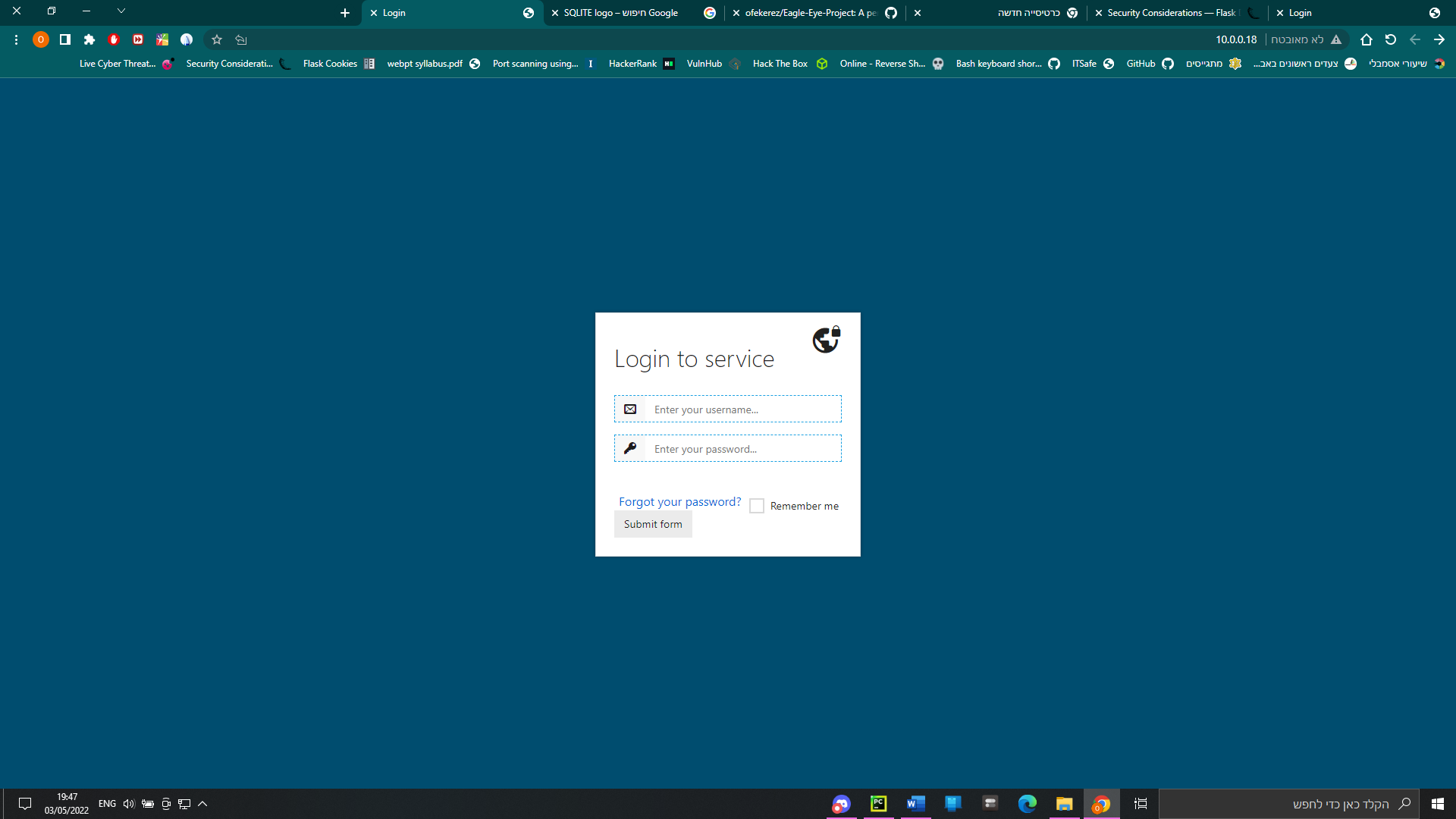
* הסבר על התהליך/יכולת
* תרשים זרימה (במקרה של תהליך)
* צילומי מסך מלווים בהסברים בשפה פשוטה

במידה ויש יותר ממשתמש אחד למערכת , לבנות פרקים שונים לדוגמא: מדריך למשתמש קצה, מדריך למנהל מערכת וכו'

* + - 1. הרשמה ראשונית למערכת.



2. התחברות משתמש קיים למערכת (Log In).



3. אימות משתמש דרך מייל.

תמונה שמכילה טקסט, שולחן

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט

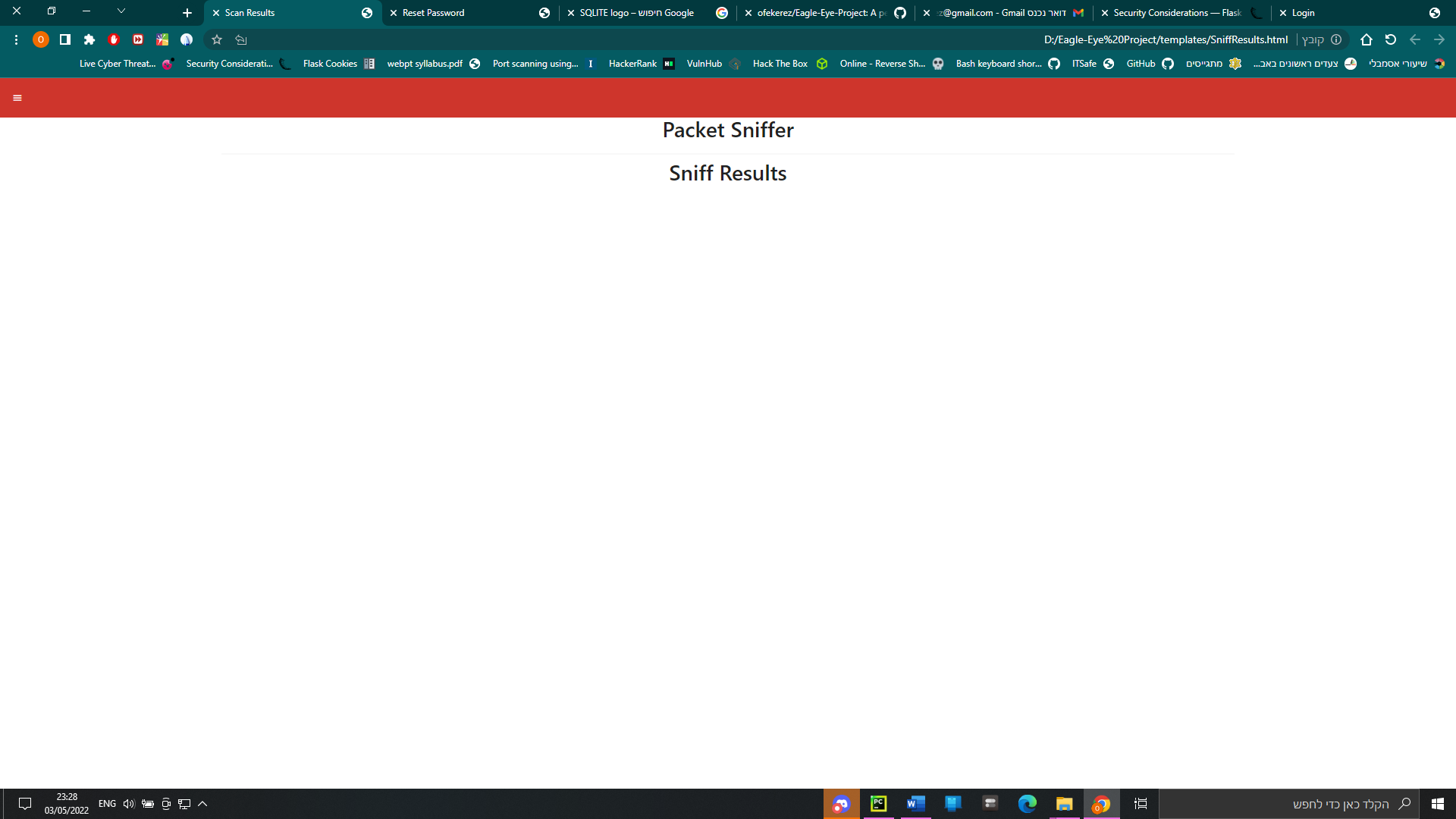
התיאור נוצר באופן אוטומטי

4. אפשרות איפוס סיסמה.

תמונה שמכילה טקסט, שולחן

התיאור נוצר באופן אוטומטי

5. אפשרות צפייה בסריקות קודמות.



6. אפשרות לסרוק את העמדות הפעילות ברשת המקומית.

תמונה שמכילה טקסט

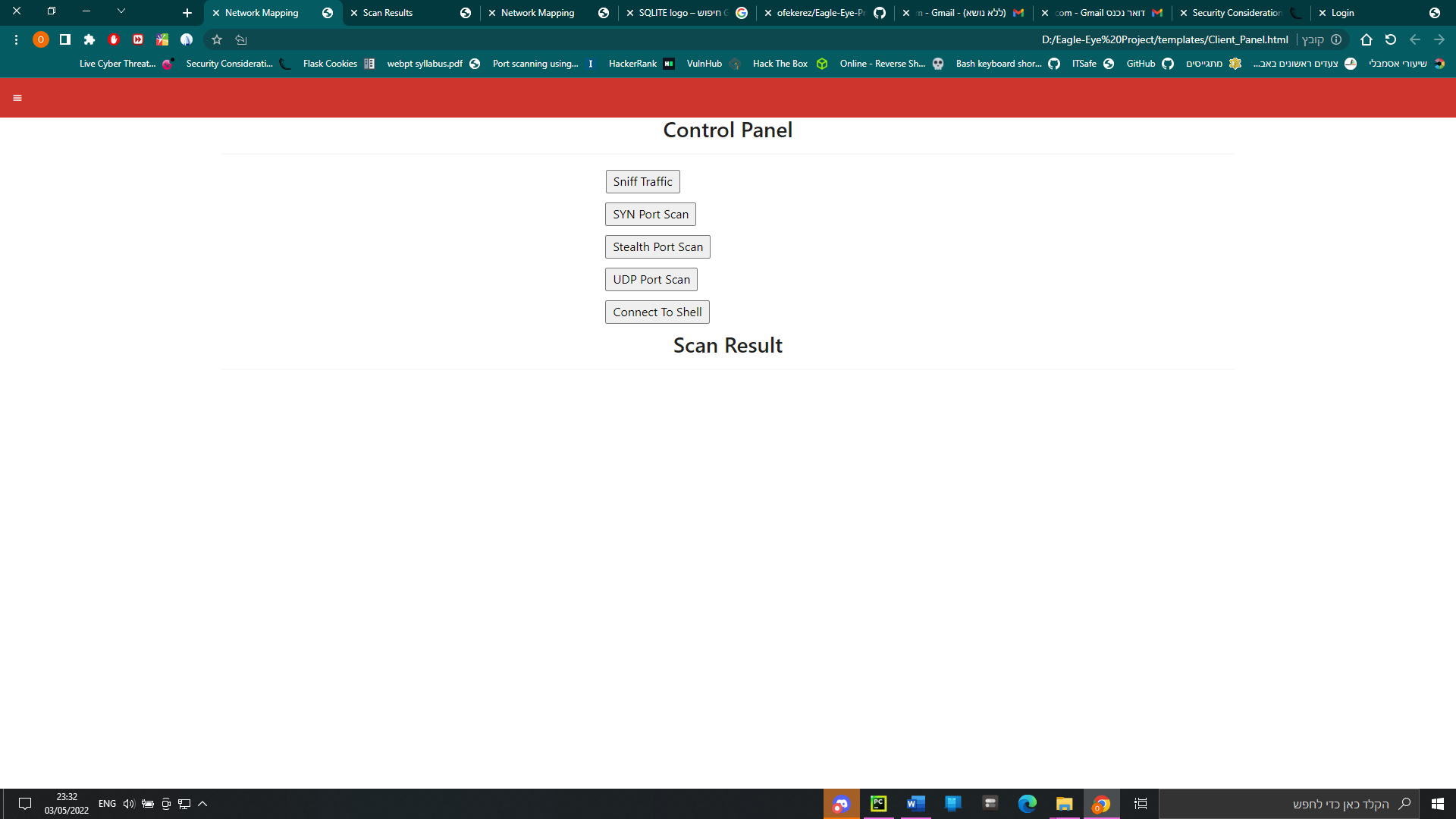
התיאור נוצר באופן אוטומטי

7. אפשרות לבצע סריקת פורטים מסוג SYN TCP על כל אחד ממחשבי הרשת.

8.  אפשרות לבצע סריקת פורטים מסוג Stealth TCP על כל אחד ממחשבי הרשת.

9. אפשרות לבצע סריקת פורטי UDP על כל אחד ממחשבי הרשת.

10. אפשרות הפעלת הסנפת תקשורת על כל אחד ממחשבי הרשת.

11. אפשרות התחברות לממשק הפקודה של כל אחד מהמחשבים באמצעות Reverse Shell. 

12. התנתקות מהמערכת( (Log out.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

**פרק ח' - 'שם המערכת' – רפלקציה**

1. מבט אישי על העבודה ועל תהליך פיתוחה:

* אתגרים שהיו לי בדרך
* אירועים מעניינים שקרו במהלך הפיתוח
* התמודדות עם קשיים, איך מה עשיתי
* הערכת הפתרון לעומת התכנון והמלצות לשיפורו
* תודות חשוב מאוד לחברים, למשפחה, למורים...

יש למלא עד עמוד ולא להתבייש בכתיבה – כאן לא המקום לצניעות.

**פרק ט' - 'שם המערכת' – ביבליוגרפיה**

1. רקע תיאורטי
2. ספרות מקצועית ספציפית לנושא העבודה (רשימת ספרים, מאמרים והפנייה לכתובות אתרים, סרטונים המכילים חומר רקע ששימש לפיתוח העבודה)
3. הערה: לא מספיק להתייחס לספרים המיועדים ללימוד שפה או כלים יישומיים ומדריכים למיניהם.
4. קישורים לאתרים באינטנרט – קישורים ל stackOverflow עם הסבר על הנושא שנחקר (לא תחביר...)

יש להקפיד להציג את הרשימה לפי כללי ה – APA ראו להלן

<https://www.oranim.ac.il/sites/heb/research-information-resource/library/consultation/writing-bib/apa/pages/default.aspx>

נספחים  
  
מסכי MVP (כולל לינק למצגת MVP)

מסכי ניהול הפרוייקט באמצעות Trello

כאן ניתן להוסיף הסברים על הטכנולוגיות שנעשה בהם שימוש, או כל מידע שיכול להועיל לקורא