

**מסמך פרויקט - תשע"ז**

**צוות:** ליעד אליהו 206975500

אלעד גהוורה 208246488

**כיתה:** יב1 עכו

**מדריך:** אלכס זקשנסקי

**מנטור:** בוריס ברזנר

# טבלת מעקב/עדכון גרסאות

|  |  |  |  |
| --- | --- | --- | --- |
| תאריך | מבצע/ת העדכון | תוכן/מהות העדכון | הסעיפים שעודכנו |
| 30/10/16 | ליעד אליהו ואלעד גהוורה | התחלת עבודה על מסמך ייזום | תיאור כללי, מטרת הפרוייקט... |
| 24.11.16 | אלעד גהוורה וליעד אליהו | פרק שני | מסמך אפיון |
| 2.1.17 | אלעד גהוורה וליעד אליהו | פרק שלישי | מסכים והמחשה ויזואלית |
| 28.1.17 | אלעד גהוורה וליעד אליהו | פרק רביעי | מסמך ארכיטקטורה  ותוכנית עבודה |
| 4.3.17 | אלעד גהוורה וליעד אליהו | שיפורים | תיקונים לדברים שהשתנו |

פרק ראשון:

מסמך ייזום

# תיאור כללי

תוכנה(כלי) לסריקה ומיפוי אתרים ומציאת חורי אבטחה ,וחולשות. הגישה לסריקת האתר ניתנת רק לבעל האתר בעזרת hash ייחודי (מנגנון הגנה).

# מטרת הפרויקט

כיום יש המון התקפות שמכוונות לאתרי web ישראליים, ולכן אנו מנסים לייצר כלי סריקה למטרת אבטחה לבעלי האתרים - כלי שסורק את דפי האתר ומנסה למצוא בהם חורי אבטחה.(כדי שיוכל לעדכן את בעל האתר על חורי האבטחה המצויים באתרו)

# סוגי משתמשים / קהל היעד של היישום

אנשים חסרי ידע נרחב בתכנות אך בעלי המוטיבציה לאבטח את האתר שלהם בעזרת התוכנה.

השימוש בתוכנה לא מסובך ואף יש הסבר על השימוש בה.

# פיצ'רים ותהליכים עיקריים

סריקה ומיפוי אתר באמצעות web crawler

הזדהות (אופציונאלית) של המשתמש בעזרת מנגנון cookies

תאימות לשרתי web שונים

סריקת חולשות בטפסי האתר

מציאת דפים המהווים חולשה

# טכנולוגיות עיקריות ושיקולים עיקריים

פיתוח למחשב, עבודה תחת מנוע פייתון עקב שימוש בשפה, רשתות ומנגנוני אבטחה.

# ידע, סיכונים ושאלות

החששות לגבי ביצוע הפרויקט הן אי הספקתו, התעמקות בנושא אחד וזניחת שאר הנושאים ו אבטחת הפרויקט.

# דרישות חומרה

-

# פתרונות קיימים

Grabber,sqlmap,nessus,vega ועוד …

פרק שני:

מסמך אפיון

# פיצ'רים ותהליכים עיקריים - הרחבה

Web crawler = wc

Vulnerability scanner = vs

● סריקת אתר באמצעות **web crawler -** השלב הראשון הוא למפות את דפי וטפסי האתר וזה התפקיד של הWC ולשם כך המשתמש נדרש להזין לממשק הWC את כתובת הURL של האתר שלו.הWC מוודא שאכן הסורק הוא בעל האתר.הWC מתחיל מהדף הראשי ונכנס לקישורים שמובילים לקישורים ובכך אוסף מידע על האתר כמו טפסים,קישורים ופרמטרים.כל תפקיד הWC הוא לאסוף כמה שיותר מידע על האתר שינוצל מאוחר יותר לסריקת החולשות בו.

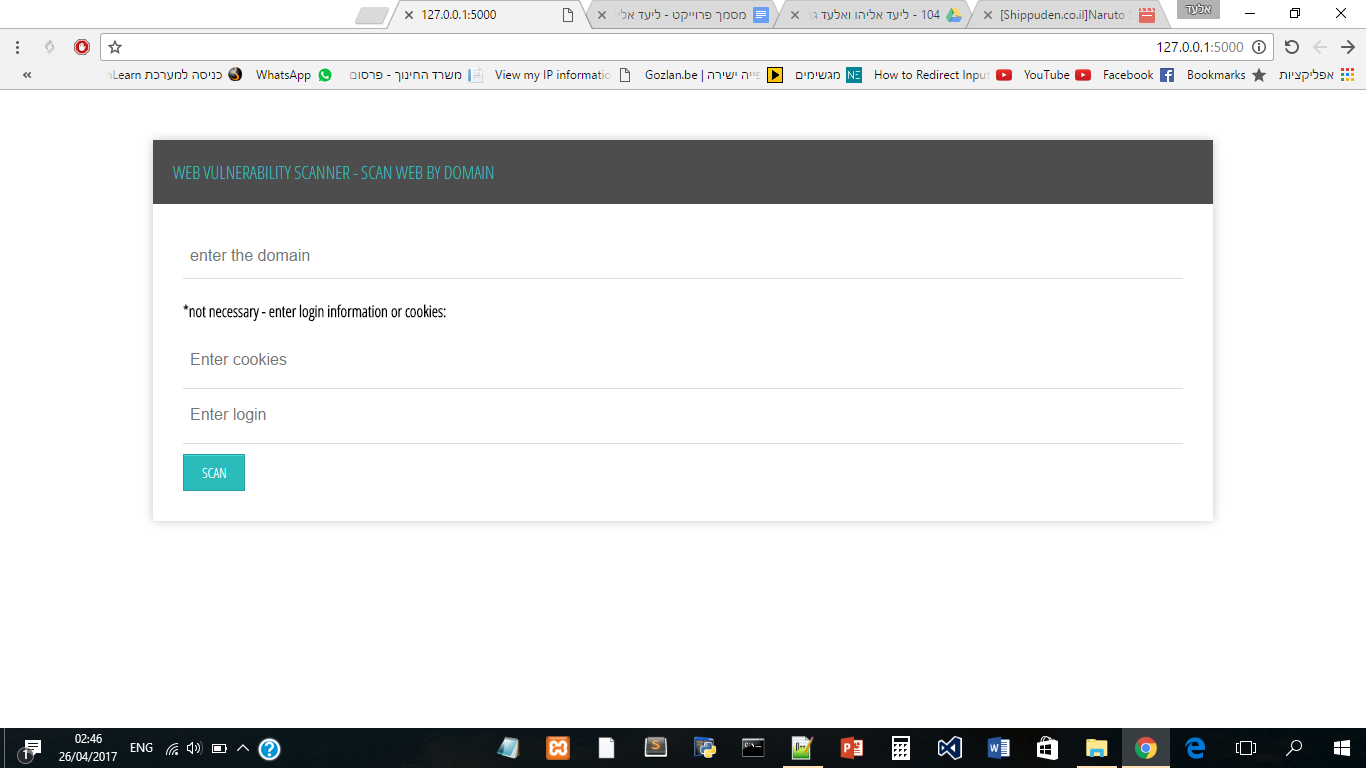
● סריקת חולשות בטפסי האתר **vulnerability scanner -** כל המידע שנאסף ע"י הWC עובר לשימוש הVS.הVS שולף ממסד מרוחק ע"י תקשורת מאובטחת חולשות נפוצות וידועות ומריץ אותם אחת אחת על הטפסים,כתובות הURL,והזרקות בפרמטרים השונים של האתר.מתקפות נתמכות הן:  
Sql injection  
Xss  
LFI/RFI  
Command exec

● תאימות לשרתי **web** שונים **-** התוכנה מתאימה את עצמה לכלל שרתי הweb האפשריים - apache , IIS , GWS.

● שימוש בהתחברות למשתמש בעזרת מנגנון **cookies -** המשתמש מזין לתוך קובץ את ערכי הcookies שאיתם ירצה להזדהות (לפי פרוטוקול מוסבר ומתועד) באתר ואיתם התוכנה מפרסרת את קובץ הcookies ומשתמשת בהם לכל אורך הsession.

פרק שלישי:

**מסכים והמחשה ויזואלית**

**המסך הראשוני - הכנסת domain האתר:**

בעל האתר יכניס את הdomain של האתר. לאחר לחיצה על הכפתור **scan**,תתחיל סריקת האתר(במילים אחרות הweb crawler וה-Vulnerability scanner ירוצו) ובנוסף יהיה מעבר למסך השני - מסך התוצאות.

בנוסף, יש אפשרות להכנסת קוקיס ופרטי התחברות למעוניינים לסריקה ייחודית יותר.(כמובן שיש מדריך באיזה פורמט לעשות זאת).

# **המסך השני - מסך התוצאות:**

מסך זה מתעדכן בצורה אוטומטית ובו נמצאים תוצאות הסריקה.

במסך זה מספר טבלאות:

1. הלינק של הdomain - הלינק של דף הבית של האתר.
2. רשימת לינקים של דפי האתר - הweb crawler שממפה את האתר, מוצא את כל הלינקים של הדפים באתר.
3. רשימת לינקים של טפסי האתר - הweb crawler שממפה את האתר, מוצא את כל הטפסים באתר.
4. רשימה של הלינקים הפריצים - לאחר שנמצאו כל הדפים והטפסים של האתר, נמצאים כל הדפים הפריצים באתר באמצעות הvulnarbility scanner, ששולף ממסד מרוחק ע"י תקשורת מאובטחת חולשות נפוצות וידועות ומריץ אותם אחת אחת על הטפסים,כתובות הURL,והזרקות בפרמטרים השונים של האתר. יש לציין שברשימה זו יצוין גם הלינק וגם סוג המתקפה (xss , sql injection , command exec , lfi , rfi).
5. לינקים מומלצים לקריאה על המתקפות.



# **דרישות שאינן פונקציונליות**

● התמודדת עם המודלריות של אתרים שונים.

● טיפול בתקלות ובפרצות אבטחה אפשריות.

● תאימות לשרתי WEB שונים.

●טיפול בשגיאות הנזרקות לתוכנה במקרי קצה.

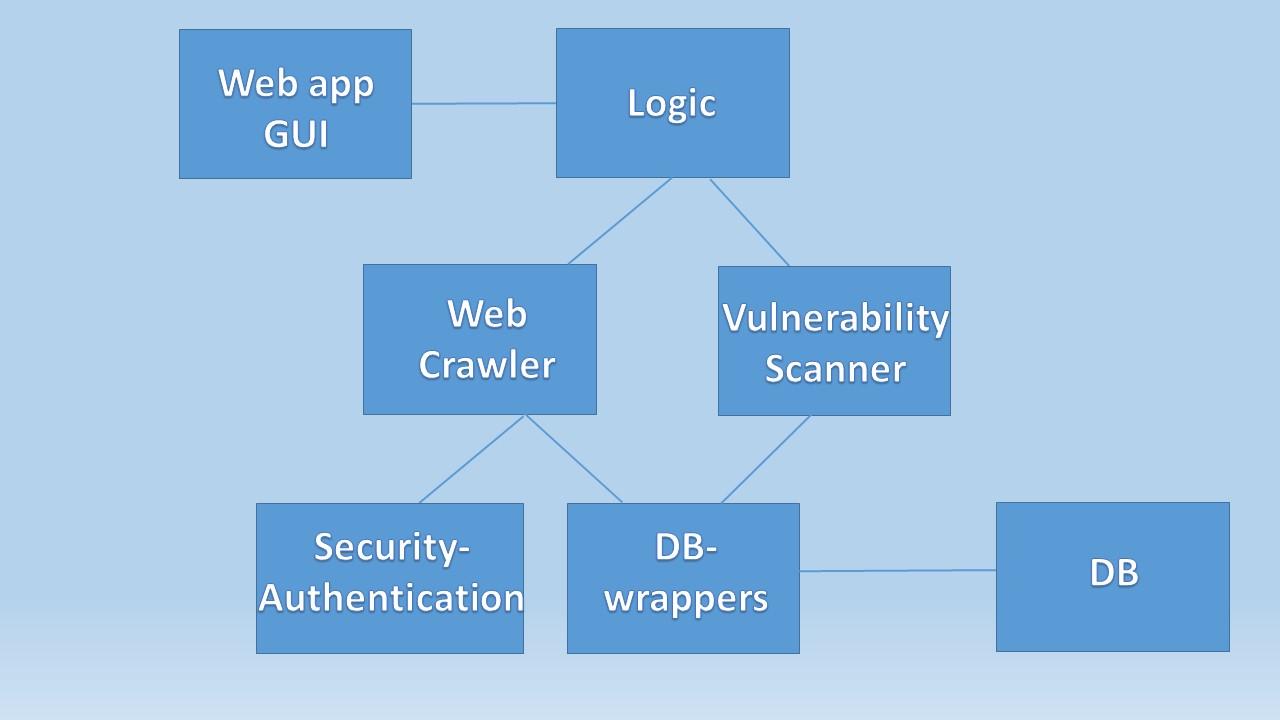
●בדיקות והצעות לשיפור החולשה באתר וסיפוק מידע על החולשה ללקוח.

●תחזור המסד לאורך זמן.

פרק רביעי:

מסמך ארכיטקטורה

Client:



GUI - המשתמש יכנס לאתר שאותו הכנו ועיצבנו, ויכניס את כתובת הURL של האתר. כאשר ילחץ על scan השרת יתחיל לרוץ ואיתו גם הWC והVS. בנוסף, יועבר המשתמש לדף 'תוצאות' שיתעדכן בזמן אמת בו יוכל לראות את הלינקים והטפסים שנמצאו באתר ע"י הWC, ואת הלינקים/טפסים הפריצים שנמצאו באתר ע"י הVS - יציין איזה סוג מתקפה ואת הלינק לדף.

Security רכיב אבטחה - אחראי על ווידוי ואימות האם מפעיל התוכנה הוא בעל האתר הנסרק.

Logic - לנהל את רצף התוכנית(שליחת פרמטרים לweb crawler קבלת הטפסים מהweb crawler ושליחתם לvulnerability scanner כדי למצוא חורי אבטחה) ובלוגיקה ולטפל בשגיאות.

Web crawler - מתחיל מהדף הראשי ונכנס לקישורים שמובילים לקישורים ובכך אוסף מידע על האתר כמו טפסים,קישורים ופרמטרים.כל תפקיד הWC הוא לאסוף כמה שיותר מידע על האתר שינוצל מאוחר יותר לסריקת החולשות בו.

Vulnerability scanner - כל המידע שנאסף ע"י הWeb crawler עובר לשימוש הVulnerability scanner.Vulnerability scanner שולף ממסד מרוחק ע"י תקשורת מאובטחת חולשות נפוצות וידועות ומריץ אותם אחת אחת על הטפסים,כתובות הURL,והזרקות בפרמטרים השונים של האתר.מתקפות נתמכות הן:  
Sql injection  
Xss  
LFI/RFI  
Command exec

DB Wrapper - מעטפת גישה לDB אשר תנהל הרשאות וגישות למערכת ותדאג שלא ירוצו פקודות לא רצוניות.

DB - רכיב מסד נתונים שמכיל טבלאות בנושאים שונים כמו cheat sheets ו דפים חלשים באתר.

**תוכנית עבודה**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סדר | משימה | תלויות תשתית | נימוק והערות | רכיב רלוונטי |
| 1 | קלט פרמטרים מהמשתמש | אין | קולט פרמטרים מבעל האתר כגון URL , הcookies בהם משתמש. | GUI |
| 2 | ניהול סריקת האתר | תלוי קלט מהמשתמש(URL)  DB | מנהל את פונקצית 'סריקת דף אתר'- סורק את כל מכלול דפי האתר ע"י שימוש בפונקציה 'סריקת דף אתר'. | Web crawler |
| 3 | סריקת דף אתר | ניהול סריקת אתר | ע"י קבלת URL/page סורקת הפונקציה את דף האתר ומוצאות בו את הפרמטרים הנחוצים - טפסים וכתובות נוספות. | Web crawler |
| 4 | הקמת DB | אין | מסד מסוג mongo וניהול הרשאות משתמשים | DB |
| 5 | יצירת טבלאות תוכן בDB | DB | טבלאות דפים חלשים, חולשות קוד, דפי ניהול. | DB |
| 6 | פונקציות שליפה מהמסד | DB | שליפת מידע דרוש מהמסד | DB wrapper |
| 7 | ניהול סריקת חולשות | Web crawler, DB | משתמש במחלקות הטיפול השונות לכל דף ובודק באיזה דפים יש חולשות(עפ"י מחלקות הטיפול) | Vulnerability scanner |
| 8 | מחלקת טיפול - XSS | Vulnerability scanner , DB | באמצעות טפסים וקישורים אותם סורקת עם cheat sheets מהDB ומדווחת על הסטטוס. | Vulnerability scanner |
| 9 | מחלקת טיפול - LFI | Vulnerability scanner , DB | באמצעות טפסים וקישורים אותם סורקת עם cheat sheets מהDB ומדווחת על הסטטוס. | Vulnerability scanner |
| 10 | מחלקת טיפול -RFI | Vulnerability scanner , DB | באמצעות טפסים וקישורים אותם סורקת עם cheat sheets מהDB ומדווחת על הסטטוס. | Vulnerability scanner |
| 11 | מחלקת טיפול - SQL injection | Vulnerability scanner , DB | באמצעות טפסים וקישורים אותם סורקת עם cheat sheets מהDB ומדווחת על הסטטוס. | Vulnerability scanner |
| 12 | מחלקת טיפול - command exec | Vulnerability scanner , DB | באמצעות טפסים וקישורים אותם סורקת עם cheat sheets מהDB ומדווחת על הסטטוס. | Vulnerability scanner |
| 13 | מנגון הגנה על האתר | הפרמטרים שמתקבלים ע"י המשתמש. | מנגנון שבודק שהאתר הנסרק באמת ע"י בעל האתר האמיתי | logic |
| 14 | main | Web crawler, Vulnerability scanner | מנהל את כל הקוד - קבלת הפרמטרים מהWeb crawler וקבלת הVulnerabilitys מהVulnerability scanner | logic |

**פרק 5: עיצוב - מדי ספרינט**

# **ספרינט 1 - web crawler**

תקציר, הסבר ותוצר: קבלת URL מהמשתמש, סריקת האתר(לפי הURL) ומציאת כל הפרמטרים הנחוצים - טפסים כתובת נוספות(כדי לסרוק את כל דפי האתר). בנוסף פונקציית ניהול לסריקה - מפעילה את סריקת הדף על כל הדפים.

תכולות טכנולוגיות: באופן הויזואלי CLI אליו המשתמש יכניס פרמטרים (URL של דף האתר). במהלך סריקת האתר יופיעו הודעות debugging.

# **ספרינט 2 - DB והחלפת הGUI מCLI לאתר:**

תקציר, הסבר ותוצר: החלפה מראה הGUI - בניית אתר בעל שני מסכים, מסך ראשון בוא יוכנס הdomain של האתר שמשם יעבור למסך השני של תוצאות סריקת הweb scanner וה-vulnarbility scanner. בנוסף בנית DB מסוג mongo, בתוכו cheat sheets של המתקפות. בין היתר בניה של DB wrapper - פונקציות שליפה מהמסד.

תכולות טכנולוגיות: באופן הויזואלי web app אליו המשתמש יכניס פרמטרים (URL של דף האתר) במעבר אל המסך השני(מסך התוצאות) יופיעו לינקים לכל דפי האתר ולכל טפסי האתר.

# **ספרינט 3 - הוספת המתקפות xss ו- sql injection:**

תקציר, הסבר ותוצר: ע"י שימוש של cheat sheets מהDB בניית 2 מחלקות טיפול "שיזריקו" לפרמטרים המתאימים את הcheat sheets מהDB ובכך יגלו את הדפים הלא מאובטחים באתר.(רק xss ו- sql injection)

תכולות טכנולוגיות:.אופן הויזואלי web app אליו המשתמש יכניס פרמטרים (URL של דף האתר) במעבר אל המסך השני(מסך התוצאות) יופיעו לינקים לכל דפי האתר ולכל טפסי האתר והלינקים לדפים והטפסים הפריצים למתקפות XSS ו- SQL INJECTION.

# **ספרינט 4 - הוספה: LFI/RFI ו- command exec:**

תקציר, הסבר ותוצר: ע"י שימוש של cheat sheets מהDB בניית 2 מחלקות טיפול **נוספות** "שיזריקו" לפרמטרים המתאימים את הcheat sheets מהDB ובכך יגלו את הדפים הלא מאובטחים באתר.(רק lfi/rfi ו- command exec)

תכולות טכנולוגיות:.אופן הויזואלי web app אליו המשתמש יכניס פרמטרים (URL של דף האתר) במעבר אל המסך השני(מסך התוצאות) יופיעו לינקים לכל דפי האתר ולכל טפסי האתר והלינקים לדפים והטפסים הפריצים למתקפות XSS , SQL INJECTION , LFI/RFI , command exec.

# **ספרינט 5 - מנגנון הגנה על האתר ושיפורים אחרונים:**

תקציר, הסבר ותוצר: הוספת מנגנון הגנה באמצעות Hash ייחודי , בנוסף עיצוב הweb app.

**נספחים**

Helper:

קובץ txt שמצורף יחד עם הפרויקט ומכיל את התוכן הבא:

לסריקה עם cookie ייחודי:

א. במסך הראשוני באתר היכנס לתיבת הטקסט שעליה רשום "enter cookies"

ב. רשום עפ"י הפורמט הבא:

key:value

key:value

לדוגמא - security:low

לסריקה עם דף התחברות:

א. במסך הראשוני באתר היכנס לתיבת הטקסט שעליה רשום "enter login information"

ב. רשום עפ"י הפורמט הבא:

שורה ראשונה:

POST/GET filename.txt \*\*יש לשים לב שמופרד בtab

שאר השורות:

key:value

key:value