מסמך פרויקט PT - תשע"ח

צוות: יונתן גבאי ומני קייזר

כיתה: אשדוד 6

מדריך: יעקב ויינרובסקי

מנטור:

הסבר כללי על המסמך

מסמך שילווה אתכם לשבועות הקרובים וגם בהמשך השנה.

אין ליצור עותקים של המסמך, ויש להשאירו בפורמט של גוגל-דוק! כל דרך אחרת עלולה ליצור בעיות למדריך או למנטור שלכם אשר עוקב אחר ההתקדמות בעבודה במסמך ובכלל. בזכות העבודה דרך גוגל-דוקס, יש גיבוי אוטומטי, סנכרון תוכן המסמך בין כל המשתתפים בו, ומעקב גרסאות.

בכל פעם שמשתנה או מתווסף תוכן מסוים למסמך, יש לציין זאת בתמציתיות בטבלת המעקב.

# **טבלת מעקב/עדכון גרסאות**

|  |  |  |  |
| --- | --- | --- | --- |
| תאריך | מבצע/ת העדכון | תוכן/מהות העדכון | הסעיפים שעודכנו |
| תאריך של היום | איך קוראים לך? | התחלת עבודה על מסמך ייזום | תיאור כללי, מטרת הפרוייקט... |
|  |  |  |  |
|  |  |  |  |

פרק 1: ייזום

# **מסמך שמטרתו סקירה כללי של הפרוייקט.**

יש למלא את המסמך עד למפגש הכיתתי הראשון. במפגש זה כל צוות יציג בקצרה את הפרוייקט שלו.

# **תיאור כללי**

כלי מבוסס web שמטרתו לעזור לבעלי אתרים לגלות חולשות אבטחת באתר ומציע פתרונות חסימת החורים.

# **מטרת הפרויקט**

נועד למענה נוח ונגיש לבעלי אתרים לפתירת בעיות אבטחה באתר..

# **סוגי משתמשים / קהל היעד של היישום**

בעלי אתרים.

# **פיצ'רים ותהליכים עיקריים**

* סריקה של חולשות אבטחה באותו אתר.
* הצעת פתרונות לבעיות האבטחה שנמצאו.

# 

# **טכנולוגיות עיקריות ושיקולים עיקריים**

לימודי ספריות של פייתון, ממשק על המחשב (לא וובי), אתר לגישה נוחה.

# **ידע, סיכונים ושאלות**

שיהיה קשה מידי, צריך לחקור המון בנושאים שאנחנו לא שולטים בהם, פיתוח מורכב

# **פתרונות קיימים**

Nessus

מסמך אפיון

# פיצ'רים ותהליכים עיקריים - הרחבה

* סריקה של חולשות אבטחה באתר - סריקה של כמה בעיות אבטחה (נקווה שנספיק כמה שיותר…) דוגמאות:
  + הזרקת SQL (באנגלית: SQL Injection) היא שיטה לניצול פרצת אבטחה בתוכנית מחשב בעזרת פניה אל מסד הנתונים. השם נובע מכך שהמשתמש מכניס קוד SQL לשדה קלט אליו אמורים היו להיכנס נתונים תמימים. באופן זה יכול משתמש זדוני לחרוג לחלוטין מן התבנית המקורית של השאילתה, ולגרום לה לבצע פעולה שונה מזו שיועדה לה במקור.
  + סריקת פורטים (port scan) - תהליך שבו מחשב מנסה ליזום חיבור עם כל פורט ופורט שבמחשב כלשהו, ולפי התגובה (או חוסר התגובה) מכל פורט, מסיק אילו פורטים פעילים כרגע ואילו לא.
  + מתקפת DoS - מתקפה שבה התוקף "מציף" (שולח כמות אדירה של פקטות) את השרת, עד שהשרת לא יוכל לעמוד בעומס שמשרה עליו התוקף, ויקרוס.
  + מתקפת DDoS - מתקפה שבה כמות אדירה של כתובות IP "מציפות" (שולחות כמות אדירה של פקטות) את השרת, כך שהשרת לא יוכל לעמוד בעומס שמשרים עליו התוקפים, ויקרוס.
  + פריצות XSS (אופציונאלי אם ישר זמן) - ניצול של פריצות אלו מאפשר הזרקת קוד זדוני אל אתר אינטרנט על ידי משתמש. כאשר משתמש תמים נכנס לאתר האינטרנט הפרוץ, מחשבו של המשתמש התמים מתייחס לקוד הזדוני כאל קוד לגיטימי של האתר הפרוץ, ומריץ את הקוד הזה. מכיוון שהקוד הזדוני נחשב לקוד לגיטימי של האתר הפרוץ, הוא יכול לקבל כל מידע הרלוונטי לאתר זה, כגון הרשאות כניסה של המשתמש לאתר. במקרים קיצוניים, למשל כאשר האתר נמצא ברשימת האתרים שהמשתמש התמים סומך עליהם, ניתן גם לנצל את הפרצה להרצת קוד זדוני בהרשאות של המשתמש עצמו.
* הצעת פתרונות לבעיות האבטחה שנמצאו - בהתחלה נפנה את הלקוח לקריאה על בעיית האבטחה, ובשלב מתקדם יותר ניתן מענה יותר יעיל ללקוח

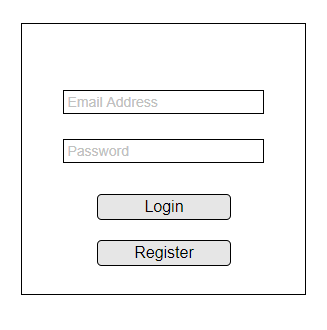
# 

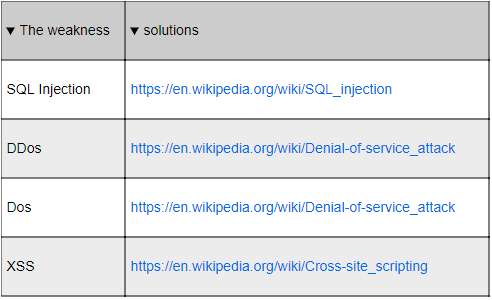
# 

# מסכים והמחשה ויזואלית

הממשק הוא אתר web שמקבל מהמשתמש את המידע הרלוונטי - כתובת אתר - ושולח אותו לשרת שעורך בדיקות על אותו אתר.

כך נראה ויזואלית המסך הראשי באתר.



# דרישות שאינן פונקציונליות

תקשורת מאובטחת (כי אפשר)

חיבור אינטרנט (יציב)

פרק ארכיטקטורה

# **מבט על**

שרטוט:



הסבר מילולי:

* תוכנת Site (אתר):
  + User Interface - ממשק למשתמש שמציג לו את האתר ואת כל האפשרויות שהוא מציע.
  + Logic - רכיב לוגיקה שאחראי לבדוק את התקינות של קלט המשתמש ושולח פקודות ל DB Wrapper ול Networking.
  + Networking - רכיב תקשורת שאחראי לשלוח ל Server את המידע שמגיע מהלקוח.
  + DB Warpper - רכיב בסיס נתונים שאחראי לשלוף, להוסיף ולעדכן נתונים מול בסיס הנתונים (ניתן להניח שהטבלאות כבר קיימות בו ונוצרו מראש). רכיבים אחרים (כמו לוגיקה) מתקשרים מולו על מנת לעבוד באופן מסודר, בטוח ויעיל מול בסיס הנתונים של התוכנה.
  + DB - בסיס הנתונים.
* תוכנת Server (שרת):
  + Networking רכיב תקשורת שאחראי להקשיב למידע המתקבל ממלכודות הדבש השונות (המידע המתקבל נשלח לרכיב הלוגיקה), ולשלוח אליהן פקודות במידת הצורך (בהתאם לבקשת המשתמש דרך רכיב הUI).
  + Logic רכיב לוגיקה - אחראי על עיבוד הנתונים והבקשות המתקבלים בתקשורת, ביצוע פעולות מול nmap ולהחזיר את התוצאות לNetworking.
  + nmap - ה API של התוכנה nmap שסורקת פורטים פתוחים.

# **עיצוב הנתונים ויישויות מידע**

הפרטים העיקריים הנשמרים במערכת הם מידע אודות המשתמשים אשר השתמשו במערכת, חותמת הזמן של השימוש וכמובן כתובת האתר עליו נעשה השימוש ותוצאות התוכנה.

הפרטים של שימוש כוללים: סוג ההתקפה, מועד התחלה (תאריך ושעה) וסיום, כתובת יעד.

# **טכנולוגיות עיקריות ושיקולים טכנולוגים**

* המערכת תפותח על גבי מערכת הפעלה לינוקס-קאלי , מכיוון ששם יש כלי פיתוח נוחים ומאימים יותר.
* הרכיבים של המערכת יפוחתו בפייתון, בגלל נוחות וספריות קלות לשימוש.

# **התאמה לאפיון**

* סריקה של חולשות אבטחה באותו אתר - בא לידי ביטוי בפרק זה ברכיב הלוגי שמשתמש בnmap בשרת וברכיב התקשורת עם האתר שעליו נעשה השימוש במערכת.
* הצעת פתרונות לבעיות האבטחה שנמצאו - בעזרת התוצאות ברכיב הלוגי בשרת ותצוגתם ברכיב הUI באתר.

תוכנית עבודה

לאחר שמסמך הארכיטקטורה שלכם הושלם ואושר, יש לפרוט אותו לשלבים שונים על גבי ציר הזמן.

מצפים לכם כמה חודשים של פיתוח, ולכן יש לתכנן מראש את סדר שלבי הפיתוח בצורה חכמה והולמת. לא כל הדרישות נולדו שוות, וגם לא תוכלו לפתח את כולן בבת-אחת. לכן יש להחליט מהי העדיפות של כל דרישה, ובהתאם לכך מה יהיה סדר הפיתוח שלהן.

בעת מתן עדיפות לדרישות, כלומר איזו דרישה תפותח קודם, השיקולים צריכים להיות בראש ובראשונה של תוכן - ולא של אילוצים טכניים. כלומר יש להתחיל מהרכיבים והדרישות שמהווים את עיקרי המהות של הפרויקט, ולאחר מכן לבנות סביבם את יתר המערכת. זוהי גישה של **מוצר מינימלי מתפקד** - כך שבסיומה של כל איטרציה ניתן יהיה להדגים פרויקט רץ ומתפקד (גם אם באופן מאוד חלקי ולא הכי יציב או מתוחכם). כמובן שגם כל איטרציה אמורה להציג פרויקט רץ אשר מתפקד יותר טוב או עם יותר פיצ'רים מאשר זה שהוצג באיטרציה הקודמת.

# מתן עדיפות לפיצ'רים, חלוקה לשלבים מסודרים

את סעיף זה יש למלא על בסיס פירוט הדרישות שכתבתם במסמך האפיון, ובהתאם לרכיבים של מסמך הארכיטקטורה.

מכיוון שכל דרישה בדרך כלל מורכבת ממספר שלבים (שכל אחד מהם הוא פיצ'ר בפני עצמו), יש להתייחס לכל שלב/פיצ'ר כזה בהתאם ובנפרד מהיתר.

לדוגמה:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **סדר** | **משימה** | **תלויות תשתית** | **נימוק והערות** | **רכיב רלוונטי** |
| 1 | זיהוי בעיית סריקת פורטים |  |  |  |
| 2 | זיהוי בעיית SQL injection |  |  |  |
| 3 | זיהוי בעיית XSS |  |  |  |
| 4 | זיהוי בעיית Dos |  |  |  |
| 5 | זיהוי בעיית DDos |  |  |  |

# חלוקה לאיטרציות

איטרציה 1 - זיהוי בעיית port scan

איטרציה 2 - זיהוי בעיית sql injection

איטרציה 3 - זיהוי בעיית xss

איטרציה 4 - זיהוי בעיית Dos

איטרציה 5 - זיהוי בעיית DDos