

נושאים מערכות הגנה לרשת - תוכנית עבודה

מגישים: ברק שליט 316222280 ונועה דוד 207465634

מטרת הפרויקט

בכוונתנו ליצור אפליקציה\service שמטרתו הגנת המשתמש מפני התקפות מסוג DNS Spoofing. האפליקציה מתרכזת בהגנה מפני תרחיש בו המשתמש ניגש לכתובת אתר מוכר, אך מנותב לאתר זדוני אחר בעל כתובת IP שונה שמטרתו פגיעה במשתמש (תרחישים מתוארים מטה). במקרה של נסיון גישה לאתר זדוני שמתחזה לאתר נורמטיבי, המערכת תחסום את גישת המשתמש לאתר הנ"ל ובמקום זאת, תנתב אותו לדף בטוח, בו הוא יזהר מגישה לאתר הזה, וכן יקבל הנחיות והכוונה לגבי איך להתמודד במצב של תקיפה מסוג זה (כלומר **יקבל גם העשרה לימודית** לגבי הנושא). בנוסף, האפליקציה תהיה בעלת GUI שבו יהיה ניתן לראות את כל תעבורות הרשת העדכניות של המשתמש עם הסטטוס שלהן, וכן אפשרות לייצא דו"ח מקיף יותר ובו פרטים על תקריות עבר שהמשתמש חווה.

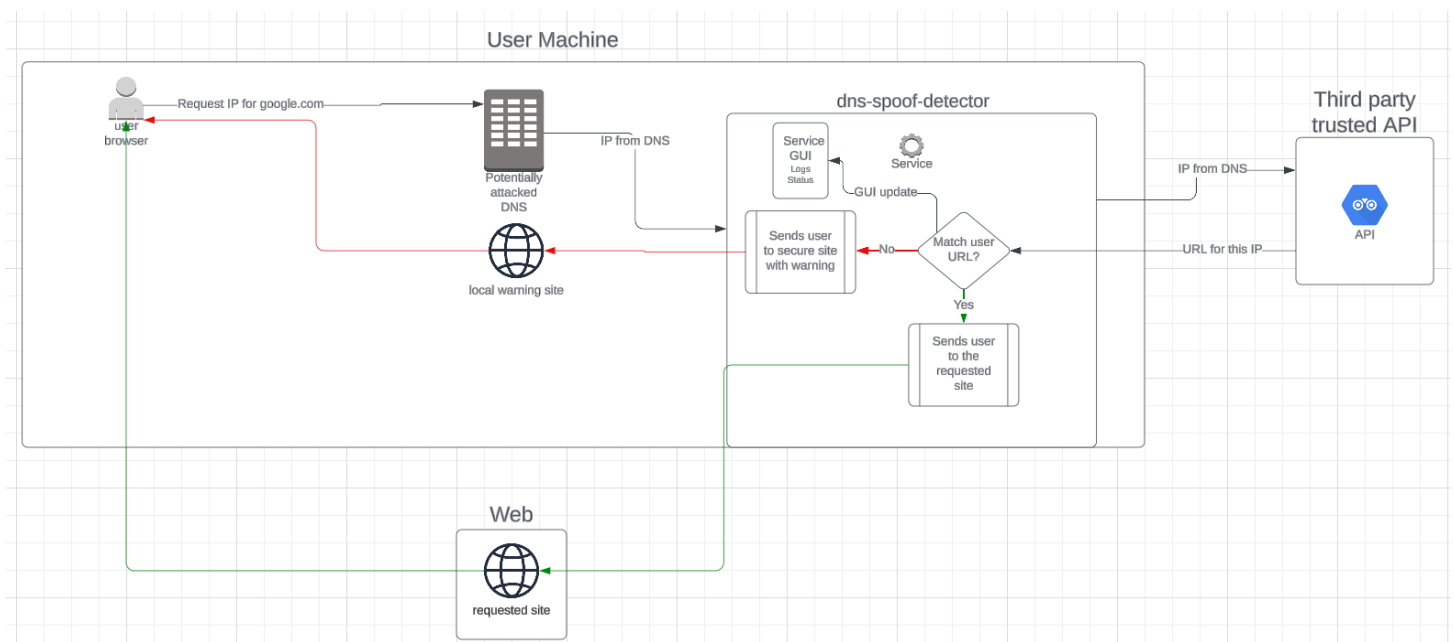
תיחום המערכת:

פרטים טכניים על הפרויקט:

1. הפרויקט נכתב בשפת פייתון.
2. הפרויקט מיועד לרוץ על מערכת הפעלה Ubuntu עם GUI (עליה מבוסס dockerfile של הפרויקט).
3. ה-API החיצוניים בהם השתמשנו בפרויקט הם: ipify, IPinfo, DNS Google API, Whois.
*יתכנו שינויים ב-API בהם נשתמש לאחר תחילת העבודה על הפרויקט כחלק מאילוצים שאנחנו עלולים להיתקל בהם.
4. לצורך הוכחת יכולות וביצוע בדיקות, נעזרנו בהקמת שרת DNS מקומי: dnsmasq.

ארכיטקטורה:

https://lucid.app/lucidchart/cf307c26-78e0-4e67-90e8-8050836f3f80/edit?viewport_lo c=358%2C-392%2C2994%2C1416%2C0_0&invitationId=inv_3a48d9bf-1b0c-4aae-a73e-f29babe8af9c



תיאור תרחישים אפשריים:

1. גניבת פרטי כרטיס אשראי:

- a. משתמש מנסה לגשת לאתר קניות ברשת (לדוג' AliExpress\Amazon וכו').
- b. מכיוון שהוא עבר מתקפת DNS, הוא מנותב במקום לאתר בעל כתובת IP אחרת שמתחזה לאתר הקניות האמיתי אליו הוא מנסה להגיע.
- c. באתר זה, המשתמש מנסה לבצע רכישה של מוצר.
- d. במהלך תהליך הרכישה, הוא מתבקש להזין פרטי כרטיס אשראי, אשר נשלחים ישירות לתוקף.
- e. המשתמש מקבל מחווה לגיטימית מהאתר שמציגה כביכול שהרכישה התבצעה בהצלחה - דבר שגורם למשתמש לא לחשוש בדבר.

2. גניבת פרטי התחברות:

- a. משתמש מנסה לגשת לרשת חברתית כלשהי.
- b. מכיוון שהוא עבר מתקפת DNS, הוא מנותב במקום לאתר בעל כתובת IP אחרת שמתחזה לאתר הרשת החברתית האמיתי אליו הוא מנסה להגיע.
- c. במהלך ניסיון ההתחברות לחשבון האישי שלו, הפרטים האישיים שלו (שם משתמש וסיסמה) נשלחים לתוקף.
- d. התוקף יכול להציג הודעת "סיסמה לא נכונה" באתר הזדוני או מחווה דומה שיכולה לגרום למשתמש למלא פרטים נוספים של חשבון המייל שלו (לאיפוס) וכן הלאה.
- e. התוקף, יכול לעשות שימושים רבים על פרטי ההתחברות של המשתמש, לרבות כופר כנגדו (שליחת בקשת כופר למשתמש תוך כדי איום פרסום פרטים אישיים של המשתמש בחשבון הרשת החברתית שלו).

3. הורדות תוכנות זדוניות:

- a. משתמש מנסה לגשת לאתר להורדת תוכן כלשהו (משחקים\מוזיקה וכו').
- b. מכיוון שהוא עבר מתקפת DNS, הוא מנותב במקום לאתר בעל כתובת IP אחרת שמתחזה לאתר ההורדות האמיתי אליו הוא מנסה להגיע.
- c. באתר זה, המשתמש מבצע הורדה של תוכן מסוים למחשב האישי שלו, אשר מתחזה להיות התוכן שהוא ביקש להוריד (משחק לדוגמה).
- d. בהרצת הקובץ שהוא הוריד, מורצת תוכנה זדונית על גבי המחשב של המשתמש וחושפת אותו לשלל סכנות.

4. איסוף מידע:

- a. משתמש מנסה לגשת לאתר חיפוש כלשהו (גוגל למשל).
- b. מכיוון שהוא עבר מתקפת DNS, הוא מנותב במקום לאתר בעל כתובת IP אחרת שמתחזה לאתר החיפוש האמיתי אליו הוא מנסה להגיע.
- c. האתר הזדוני מציג לו תוצאות אמיתיות על החיפושים שלו (מנתב אותו לתוצאות מהאתר האמיתי), ולאורך חודשים אוסף את השאליות שהמשתמש מחפש באתר הזדוני.
- d. הדבר חושף את המשתמש לשלל סכנות, בהם אפשרות של בקשות כופר, או שיווק מכוון ללא ידיעת המשתמש (אם חיפש לגבי הלוואות למשל, התוקף יכול לשווק לו הלוואות או מוצרים בסגנון זה ולנצל את המידע החסוי שיש ברשותו לגבי התעניינות המשתמש במוצרים מסוימים).

קיימים עוד תרחישים רבים שכולם סובבים סביב התחזות לאתר לגיטימי במטרה לגרום למשתמש למסור פרטים שהוא לא התכוון למסור לגורם שלישי.

אתגרים וקשיים:

1. מהנחת העבודה שה-DNS של המשתמש הותקף, נצטרך לוודא שאכן התשובה שנקבל מה-API החיצוני (שמטרתו לוודא כתובת URL אך מול IP) אמינה בעצמה, שכן תוקף יכול גם לשבש את התשובה שנקבל ממנו או להתחזות אליה.
2. כדי להתגבר על קושי זה קיימות מספר דרכים -
 - a. גישה ל-API הרצויים ישירות דרך כתובת IP ולא URL.
 - b. עבודה אל מול מספר API חיצוניים (דבר שיוריד את הסבירות שהתוקף זיהה את כולם ופגע בתקשורת עימם).
 - c. תקשורת HTTPS ושימוש ב-CERTIFICATES אל מול ה-API הנ"ל כדי להבטיח תשובה אמינה.

חוזקות:

1. האפליקציה רצה באופן תמידי ורציף על המכונה של המשתמש ולכן המשתמש תמיד מוגן מפני מתקפות DNS, מבלי שום צורך שלו לוודא כל בקשה ובקשה שהוא שולח לדפדפן.
2. לאפליקציה יש GUI ברור וידידותי שמקל על המשתמש להבין מה הסטטוס הנוכחי שלו והאם הוא נחשף לתקיפה.
3. האפליקציה מנתבת את המשתמש לחומרי קריאה אודות המתקפה שהיה חשוף אליה ולכן לא רק מגנה עליו מפניה, אלא גם מספקת לו הכוונה איך להיות מוגן בהמשך.

לוחות זמנים:

מיני פרויקט תוכנית עבודה

שבוע 1	שבוע 2	שבוע 3	שבוע 4	שבוע 5	שבוע 6	שבוע 7	שבוע 8	שבוע 9
מחקר ומתכנן	מחקר לגבי נושא לעבודה							
	כתיבת תוכנית עבודה							
	מחקר לגבי שפת קוד לאפליקציה							
	מחקר לגבי API חיצוניים							
	מחקר לגבי סביבת עבודה							
	מחקר לגבי ספריות עזר							
	מחקר לגבי תהליך הבדיקות							
הכנת סביבה			הקמת סביבה וכתיבת Docker file					
			התקנת שפת קוד נבחרת					
			התקנת ספריות נדרשות					
			התקנת DNS לוקאלי					
כתיבת קוד			פיתוח קומפוננטה A					
			פיתוח קומפוננטה B					
			פיתוח קומפוננטה C					
			פיתוח קומפוננטה D					
			פיתוח קומפוננטה E					
			פיתוח קומפוננטה F					
בדיקות							בדיקות	