עבודת מחקר בנושא:

אבטחת אתרים

שם המגיש: אריאל קמחי

מוגש למרצה: רעיה חזי

אתר אינטרנט מוגדר כאוסף של דפים המקושרים ביניהם, ונמצאים לרוב על אותו שרת. כמו כן דפים אלו עוסקים ברוב המקרים בנושא המשותף לכולם. נציין כי דף הינו יחידת מידע שמתפקדת כמו קובץ מלל שרושמים על מחשב, ונשמר עליו במקום מיוחד, ואילו למושג שרת יש שתי משמעויות הקשורות זו לזו. משמעות אחת שניתן לייחס לשרת היא תוכנת מחשב המעניקה שירותים שונים לתוכנות לקוח. בנוסף, ניתן להתייחס לשרת כאל מחשב פיזי שמריץ תוכנות מחשב שונות, ודרכן מספק שירותים שונים למחשבים אחרים. כדי ששרת יתפקד כראוי לאורך זמן קיימות עבורו דרישות חומרה גבוהות יותר מאשר מחשב "רגיל" שאינו מתפקד כשרת.

אתר אינטרנט מורכב מתכנים שונים, סטטיים, עיצוביים ודינמיים, הכתובים באמצעות קוד, לרוב בשפות HTML או XHTML, CSS ו-JavaScript או VBscript בהתאמה. תכנים אלו מוצגים באמצעות דפדפן, תוכנה שמתרגמת אותם לתצוגה גרפית בדפים השונים באתר. נוסף על כך, תוכנה זו מאפשרת לעבור בין דפי האינטרנט השונים, כלומר לגלוש באתר. בגלישה זו המשתמש מספק לדפדפן את הכתובת הרצויה אליה הוא מעוניין להגיע. את כתובת זו ניתן לספק בשתי דרכים עיקריות. האחת באמצעות שם האתר עצמו, והשנייה באמצעות כתובת ה-IP שלו. אם המשתמש מספק את שם האתר עצמו אז הדפדפן מתרגם אותו לכתובת IP באמצעות שרת DNS. מכאן ניתן להסיק כי הדרך הראשונה היא דרך עקיפה, והשנייה ישירה. כמו כן ניתן לגשת לאתרים הרצויים בדרכים עקיפות נוספות כגון מנועי חיפוש כמו Google, Yahoo!, Bing או באמצעות קישור ספציפי אליהם. הקישור יכול להוביל לאתרים אלה דרך כפתור, כותרת ראשית, תמונה, סרגלי ניווט ועוד. נציין כי לכל אתר לרוב קיים דף מרכזי הנקרא דף הבית של האתר, וממנו ניתן באופן ישיר או עקיף לגשת לאוסף הדפים שלו וכן לאתרי אינטרנט חיצוניים שונים. נדגיש כי דרכים אלה עוזרות למשתמשים לנווט ולהתמצא באתרים השונים בצורה נוחה, פשוטה ומהירה לעומת שתי הדרכים הראשונות המצוינות לעיל.

כמו כן כל כתובת של דף באתר מוצגת במבנה הנקרא URL, שמשמעותו היא Uniform Resource Locator, ובעברית מען משאבים אחיד. לפי מבנה זה כל כתובת מורכבת מכמה חלקים:
החלק הראשון הינו הפרוטוקול שלפיו ניתן לגשת לאתר, למשל HTTP, HTTPS. החלק השני נקרא שם תחום משני (באנגלית Subdomain), למשל WWW, כלומר World Wide Web או בתרגום לעברית "רשת כלל-עולמית". החלק השלישי נקרא שם התחום (Domain), שהוא השם הייחודי של אתר, המבדיל אותו משאר האתרים המצויים ברשת האינטרנט. דוגמאות לשמות כאלה הם Google, Walla, Facebook, Tweeter, Ynet ועוד. חלק נוסף שממנו מורכבת כתובת של אתר נקרא Top Level Domain או בקיצור TLD, ובעברית סיומת אינטרנט. לסיומות אלה יכולות להיות משמעויות שונות. סיומות כאלה יכולות להיות גנריות או לייצג שם של מדינה. למשל הסיומת com. מייצגת מוסדות מסחריים-בינלאומיים, ואילו הסיומת org. מייצגת מוסדות ללא מטרות רווח. לעומתן הסיומות il., us., uk. מייצגות את המדינות ישראל, ארצות הברית ואת בריטניה בהתאמה. כמו כן לאחר הסיומת של כתובת דף באתר לעיתים מופיע חלק נוסף שמשמש לניתוב פנימי הקשור לשרת. אם האתר משתמש בשאילתה שנועדת לחיפוש בתוך האתר, אזי לאחר ניתוב זה יופיע סימן שאלה. כמו כן אחריו יופיעו הפרמטרים של כל בקשת חיפוש אשר מוצגים במבנה מיוחד. ברוב המקרים מבנה זה מורכב ממפתח הבקשה, סימן "=" ולאחר מכן הערך שאותו מעוניינים לחפש. כמו כן בין כל ערך כזה מפריד סימן "&". למשל אם מחפשים באתר מסויים את הביטוי "Top Ten" אז בחלק הנוסף יופיע תחילה הניתוב הפנימי, לאחר מכן סימן שאלה, ולבסוף מפתח, סימן "=", המילה Top, סימן "&" ואחריו מפתח נוסף, סימן "=" והמילה Ten.

אולם, בגלישה בדפי האינטרנט השונים המשתמשים ומפתחי האתרים עלולים להיחשף לסכנות שונות. לפי אתר Owasp, נכון לשנת 2017 קיימים עשרה איומים קריטיים הקשורים לדפים אלה. האיום הראשון הינו הזרקת קוד זדוני (Injection), והשני נקרא הזדהות שבורה. כמו כן, האיום השלישי קשור לחשיפת מידע רגיש, ואילו הרביעי עוסק בישויות XML חיצוניות (XXE). האיום החמישי נקרא גישה שבורה, ולעומתו השישי עוסק בניהול תצורה לא מאובטח. האיום השביעי הוא XSS, הידוע גם כ- Cross Site Scripting. האיום השמיני נקרא פתיחה לא מאובטחת של רצף סדרתי, ואילו התשיעי קשור לשימוש ברכיבים עם פגיעויות ידועות. האיום הקריטי העשירי והאחרון לפי אתר Owasp עוסק בתיעוד וניטור בלתי מספקים. כל אלה עלולים לשבש או להשבית את התפקוד הנורמטיבי של אתר, ואף להסב לו ולמשתמשיו נזק משמעותי.

האיום הראשון, הזרקת קוד זדוני, נחשב לאיום הכי משמעותי מבין איומים אלה. איום זה בא לידי ביטוי כאשר תוקף שולח קוד עוין המכיל שאילתה או פקודה לרכיב מסויים באתר שמתרגם

אותן. קוד כזה עלול להטעות את הרכיב, ואף עלול לגרום להשלכות חמורות כגון הפעלת פקודות שאינן רצויות, איסוף מידע הקשור לאתר באופן בלתי חוקי, אובדן או השחתה של מידע, התקפת מניעת שירות וכדומה. קוד זדוני שתוקפים עלולים להזריק לאתרים השונים נכתב לרוב בשפות שאילתה כגון SQL, NoSQL, באמצעות פקודות מערכת ההפעלה, באמצעות פרוטוקול תקשורת הנקרא LDAP, מתרגמי XML וכו'. נדגיש כי הזרקה כזאת מתאפשרת כיוון שרוב האתרים כיום מקושרים לשרת נתונים. הזרקת קוד הנחשבת למסוכנת ביותר מבין ההזרקות שלעיל הינה הזרקת SQL (SQL Injection).

הזרקת SQL הינה יישום אבטחה המאפשר לתוקפים להכניס פקודות ושאילתות SQL שונות לרוב לתוך אובייקטים המקבלים קלט ממשתמש. הזרקה כזאת כאמור עלולה לגרום להם לגשת למידע בצורה בלתי חוקית, ואף להפעיל עליו מניפולציות שונות כגון מחיקת המידע, שינוי המידע, הוספת מידע שאינו רלוונטי וכדומה. כמו כן, התוקפים עלולים לשנות את פעילות שרת הנתונים שאליו הם ניגשים באמצעות הזרקה זו. החולשה של דפים באתרי האינטרנט השונים למעשה מתבטאת ביכולת הרכיבים שלהם לזהות פקודות ושאילתות SQL כחוקיות (כולל את אלה שבפועל אינן רצויות).

כיום קיימות ארבע דרכי פעולה מרכזיות על מנת להגן על אותם אתרים מפני הזרקת SQL. דרך ראשונה היא שימוש בפקודות מוכנות ושאילתות עם פרמטרים. דרך שנייה היא שימוש בפרוצדורות (Stored Procedures). דרך נוספת נקראת רשימה "לבנה" של בדיקת קלט מהמשתמש (Whitelist Input Validation). דרך הפעולה הרביעית קשורה גם היא לקלט מהמשתמש ונקראת "בריחה מכל מה שהמשתמש מספק כקלט", ובאנגלית: Escaping All User Supplied Input.

לפי הדרך הראשונה השימוש בשאילתות עם פרמטרים גורם למפתח האתר, כלומר למי שבונה אותו להגדיר תחילה את כל מרכיבי הקוד שקשור ל-SQL ובו הוא משתמש בשרת הנתונים הקשור לאתר, ולאחר מכן להעביר את הפרמטרים הדרושים לכל שאילתה. עיצוב קוד בצורה זו מאפשר לשרת להבחין בין קטע קוד לבין מידע ללא קשר למה שמשתמש מכניס כקלט. כמו כן באמצעות שימוש בפקודות מוכנות ניתן לוודא ברוב המקרים שתוקף לא יוכל להכניס פקודות SQL אשר עלולות לשנות את תכני השאילתות שכתב המפתח.

בדומה לשימוש בשאילתות עם פרמטרים, השימוש בפרוצדורות (Stored Procedures) לרוב מאפשר הגדרה של מרכיבי הקוד והעברה של פרמטרים באופן הרצוי. יתרון נוסף שיש לשימוש בפרוצדורות הוא אחסון קטעי הקוד בזיכרון של שרת הנתונים עצמו. נדגיש כי בשתי הדרכים ניתן למנוע הזרקת SQL, והן נחשבות ליעילות באותה המידה. נשים לב כי בפרוצדורות משתמשים בדרך כלל בפקודות ושאילתות אשר שומרות על שלמות המידע בשרת הנתונים ואינן משנות אותו. בנוסף, אם נעשה שימוש בפקודות ושאילתות דינמיות, כלומר כאלה אשר משנות את המידע אז מפתחי האתרים השונים ישתמשו בבדיקות תקינות הנקראות גם Validation Checks ואו בכלים הקשורים לדרך הרביעית המפורטת מטה.

נוסף על כך, באמצעות רשימה "לבנה" של בדיקת קלט ניתן לוודא אם משתמש מחפש עבור המידע הרצוי. נציין כי בשרתים המשתמשים ב-SQL עובדים עם טבלאות, ובכאלה המשתמשים ב-NoSQL עובדים עם קבצי JSON שבהם יש מפתחות וערכים על מנת לשמור המידע הרלוונטי. לכן, מיפוי של שמות הטבלאות והעמודות או המפתחות והערכים בסדר מסויים (עולה או יורד) מונע מהתוקף להשתמש בפקודות ושאילתות אשר עלולות לפגוע במידע השמור בתוכם. נציין כי ניתן להמיר את תוכן הקלט המתקבל מהשתמש לסוג אחר השונה ממחרוזת. מחרוזת הינה אוסף של תווים התחומים על ידי סוגים שונים של מירכאות. דוגמאות לסוגים השונים ממחרוזת הם מספר שלם, מספר עשרוני, ערך בוליאני או תאריך. המרת מחרוזות לסוגים אלה מונעת מהתוקף לשרשר אותן ולהרכיב שאילתה או פקודה שעלולה לשנות את המידע הנ"ל.

הדרך הרביעית, "בריחה מכל מה שהמשתמש מספק כקלט" עובדת באופן הבא:
בכל מערכת לניהול מסד נתונים משתמשים בתרשימים עם ערכים שבהם שמורות שאילתות מסוימות. באמצעות שימוש בתרשימים אלה ניתן במערכת זו להבחין בין קוד שכתב מפתח האתר לקלט שהכניס המשתמש, ולפיכך ניתן למנוע הזרקת SQL.

דרך פעולה משנית נוספת שניתן לנקוט בה כדי לספק הגנה לאתרים מהזרקת SQL קשורה להרשאות, וזאת על מנת למזער את הנזקים העלולים להיגרם ממנה. דרישות מינימאליות של הרשאות לביצוע פעולות מסוימות כגון האפשרות להתחבר כמנהל מערכת (Administrator),

והאפשרות לערוך נתונים, לקרוא או להריץ אותם משפיעות בצורה משמעותית על שלמות ותקינות המידע של אתר. נשים לב כי אם משתמש למשל מצליח להתחבר כמנהל מערכת אז יהיו פתוחות בפניו כל ההרשאות האפשריות למניפולציות שניתן לעשות על המידע הקיים באתר, ולכן זה מצב שנחשב מסוכן. באותה מידה אם היו פתוחות בפני משתמש מסויים הרשאות שלא היו אמורות להיפתח בפניו כמו עריכת מידע אז אותו משתמש יוכל לנצל זאת לטובתו והוא עלול לפגום במידע הקיים באתר.

לכן נקיטה בדרכים אלה מספקת אמצעי הגנה ומניעה מהזרקת SQL. אולם, למרות אמצעים אלו, נכון לשנת 2017 הזרקת SQL נחשבת לאיום המסוכן ביותר על אתרי האינטרנט.

איום אבטחה נוסף נקרא XSS, הידוע גם כ- Cross Site Scripting. בדומה לאיום הקודם גם באיום זה יש שימוש בהזרקת קוד, אך בשונה ממנו הקוד כתוב בשפת JavaScript, וניתן להכניסו הן לשורת החיפוש בדפדפן והן לתכנים אחרים אשר קולטים מידע מהמשתמש. כמו כן בעת הרצת הקוד התוקף עלול לגנוב מידע ממשתמשי האתר השונים, ואף לבצע בשמם פעולות מסוימות. איום אבטחה זה מתחלק לשלושה סוגים - XSS זמני (XSS Reflected), XSS קבוע (נקרא גם XSS Stored או XSS Second-order) ו-XSS מקומי הנקרא גם XSS DOM.

הסוג הראשון, XSS זמני, הנחשב לסוג הנפוץ ביותר של XSS, מתרחש כאשר באתר מסויים, הנחשב למזויף, התוקף משתמש בקוד זדוני אשר פועל לאחר כל פעם שמשתמש מכניס קלט כחלק מתשובה לבקשות מסוימות כגון חיפוש אתר במנוע חיפוש, שליחת מייל, העלאת קבצים לשרת אחסון נתונים ועוד. לכאורה, אם כל משתמש היה מבחין בקוד כזה אז הוא היה נמנע מלהגיע לאותו אתר. מצד שני, התוקף יכול לשכנע את המשתמשים אשר גולשים דרך הדפדפן להיכנס לאתר המבוקש, ובכך לאסוף עליהם מידע רגיש.

בדומה לסוג הראשון, גם ב-XSS קבוע, אשר מהווה את הסוג המסוכן ביותר של XSS, יש שימוש בקוד זדוני הפועל לאחר קבלת קלט מהמשתמש, אלא שבניגוד לסוג הקודם מדובר על מידע שמאוחסן באופן קבוע, לרוב באמצעות שרת נתונים. דוגמה הממחישה זאת היא הכנסת הודעה בפורום או באתר שניתן לפרסם בו הודעות כמו טוויטר, המכילה קוד זדוני, ולכן המשתמשים הנחשפים להודעה זו עלולים לחוות התקפה מסוג זה. זאת ועוד, התוקף המנצל פרצת אבטחה זו עלול לגנוב מידע כגון נתוני אימות של המשתמשים הצופים בהודעה זו, ואף להתחזות אליהם ולפרסם בשמם הודעות שלא התכוונו לפרסם.

לעומתם ב-XSS מקומי, הנקרא גם XSS DOM, התוקף משתמש באתר קיים ומנצל פרצת אבטחה בו על מנת לאסוף מידע מהמשתמש, כלומר האתר מוצג למשתמש כאתר מוכר וכביכול מאובטח, אך הכתובת שלו למעשה מובילה אל אותו המחשב ממנו יצר התוקף את האתר אשר בו הכניס קוד זדוני למטרה זו. כאמור גם בדרך זו איסוף המידע מתבצע באמצעות קבלת קלט מהמשתמש. דוגמה לכך היא יצירת אתר אינטרנט המתחזה לאתר משרד הפנים, אך למעשה כתובת ה-URL שלו תהיה שונה מזו של האתר המקורי.

כיום קיים מודל אשר מונע התקפות XSS, בעיקר משני הסוגים הראשונים, ולפיו קיימים כללים עיקריים. נדגיש כי לפי כללים אלה חשוב להגן לא רק על תוכן האתר ועיצובו, אלא גם על הדינמיות שלו אשר באה לידי ביטוי בשפת JavaScript. הכללים הללו קשורים לבניית אתרים המורכבים מהאלמנטים השונים הקיימים בהם.

לפי הכלל הראשון יש להבחין בין מידע שניתן לסמוך עליו למידע שאינו כזה, כלומר חשוב להיזהר מהכנסת מידע שאינו מהימן לתוך אלמנטים הנמצאים באתר. למשל הכנסת URL מאתר אינטרנט חיצוני שאינו מהימן לתוך JavaScript, או הכנסת קטע קוד הכתוב בשפה זו ממקור חיצוני שאינו מהימן נחשבות לפעולות מסוכנות שאותן חשוב למנוע.

לפי הכלל השני, במקרה שהוכנס מידע לא מהימן לאלמנטים באתר יש שיטות להגן עליו. שיטה אחת היא החלפת תווים המייצגים משמעות אחת למחרוזת המייצגת את אותה משמעות. נקיטה בפעולה זו מונעת הרצה לא תקינה של אותם אלמנטים. מחרוזות שיכולות להחליף תווים המייצגים את אותה משמעות הם & במקום &, < במקום <, > במקום >.

לפי הכלל השלישי הקשור לתכונות מסוימות של אלמנטים הנמצאים באתר (גובה, רוחב, שם, ערך וכו') יש לשים לב תחילה לתווים המשלבים מספרים ואותיות, ולהשתמש בעבור ערכים שלא מכילים אותם בקוד ASCII אשר ימירם למספר המייצג אותם. תווים אשר לא מכילים אותיות ומספרים, וניתן להמירם לקוד הנ"ל הם התו רווח, התו %, התו + וכדומה.

לפי הכלל הרביעי הקשור לאלמנטים שכתובים ב-JavaScript, במקרה שמכניסים מידע שאינו מאובטח או מהימן רצוי להכניסו בצורה של מחרוזת. כמו כן אם משתמשים במידע כזה לא מומלץ להכניסו לאלמנטים שאינם בלוקי סקריפט או יומן המטפל באירועים דינאמיים שונים היכולים להתרחש באתר. פעולה כזאת נחשבת למסוכנת, וזאת מכיוון שתוקף יכול להשתמש בתווים מיוחדים כגון רווח, התו +, התו = ועוד על מנת לבצע מניפולציות שונות העלולות לפגוע בתפקוד האתר ובמשתמשיו השונים.

כמו כן קיימים ארבעה כללים נוספים עליהם לא נפרט, כיוון שהעיקרון המנחה למניעת תקיפת XSS משני הסוגים השונים המצוינים לעיל קשור לשימוש במידע מהימן ולדרכי פעולה שונות שניתן לעשות במקרים שבהם מחליטים להסתכן ולהשתמש במידע שאינו כזה.

גם עבור התקפת XSS DOM קיימים כללים שונים למניעה והגנה מפני שימוש במידע שאינו מאובטח בבניית אתרים. נזכיר כי כללים אלה מתייחסים לכל האלמנטים המרכיבים את האתרים השונים הן בשפת HTML, האחראית על התוכן עצמו, הן בשפת CSS האחראית על העיצוב והן בשפת JavaScript האחראית על הדינמיות.

לפי הכלל הראשון המתייחס לשינוי תוכן של דף באתר אינטרנט באמצעות שפת JavaScript באופן ישיר יש לכתוב את המחרוזות שבהן משתמשים בצורה מוצפנת, וזאת על מנת למנוע הרצה לא רצויה של תכונות או פונקציות מסוימות המשתמשות בהן. הצפנה כזאת יכולה למשל למנוע מתוקף לשנות את צבע הרקע של כותרת באתר באמצעות שפת JavaScript.

נדגיש כי אם קיים תוכן בדף אינטרנט מסויים אז לרוב הוא נכתב באמצעות קוד בשפת HTML, וכמו כן עיצוב התכנים נעשה באמצעות שפת CSS. אולם ניתן ליצור אלמנטים חדשים ולעצב אותם מחדש בדף גם באמצעות שפת JavaScript. לכן, לפי הכלל השני כאשר משתמשים בשפה זו כדי ליצור אלמנט שאינו מאובטח (ולא ב-HTML) יש להצפין את המחרוזות לפי הכלל הקודם, וזאת בתנאי שאין שינויים של עיצוב, של קישורים הנמצאים בדף, וכן של תכונות מסוימות שאינן קשורות ליומן המטפל באירועים אשר יכולים לקרות באתר.

כמו כן קיימים חמישה כללים נוספים הקשורים לאלמנטים שונים כגון כתובת URL, אתחול משתנים בשפת JavaScript וכדומה. העיקרון המנחה למעשה חוזר על עצמו בכל שבעת כללים אלה המיועדים להגנה ולמניעה של התקפת XSS DOM. לפי עיקרון זה יש להצפין את המחרוזות שמשתמשים בהן על מנת להכניס מידע בצורה מאובטחת לאלמנטים השונים גם כאשר הם מגיעים ממקור שאינו מהימן.

נציין כי הכללים אשר הוזכרו למניעת התקפות XSS קשורים כאמור למפתחי אתרים. בנוסף גם משתמשים אשר גולשים דרך דפדפנים באתרי האינטרנט השונים יכולים לנקוט במשנה זהירות ולשים לב להתנהגויות חשודות בהם. למשל, אם הם גולשים באתר מוכר שנחשב למאובטח כגון Google, וכתובת ה-URL שונה מהכתובת המקורית של אתר זה מומלץ לצאת מהדף במקום להשתמש בו כמנוע חיפוש לכל דבר. דוגמה נוספת להתנהגות חשודה יכולה להתבטא בקבלת הודעה עם קישור לאתר אחר שאינו מוכר דרך המייל, כלומר משתמש מסויים עלול לקבל מתוקף קישור מסויים לאתר שלו עם הודעה, ובה התוקף ינסה לשכנע את המשתמש בעזרת כלים הקשורים להנדסה חברתית שזה מגיע ממקור מאובטח ומהימן, ואולם הקישור עלול להכיל קוד זדוני שעלול לפגוע במשתמש. זאת ועוד, באתרים שבהם ניתן לפרסם הודעות כגון פייסבוק או פורומים שונים מומלץ להפעיל שיקול דעת לפני שמפרסמים מידע רגיש, כגון תעודת זהות או מספר חשבון בנק של המשתמשים, וזאת מכיוון שתוקף עלול לנצל את המידע הזה לטובתו, וכמובן לרעתם.

לסיום, כיום קיימים איומי אבטחה שונים על אתרי האינטרנט אשר משתמשים בהם בעולם כולו הן בארגונים או עסקים שונים, והן בשימוש הביתי. בין איומי האבטחה השונים ניתן למצוא את SQL Injection, Cross-Site Scripting, הזדהות שבורה, חשיפת מידע רגיש, XXE, ניהול תצורה לא מאובטח ועוד. ממה שנאמר בפסקאות הקודמות עולה כי קיים צורך הן למפתחי האתרים והן למשתמשים השונים להעלות את המודעות לסכנות השונות העלולות להיגרם בגלישה בהם. כמו כן מומלץ לנקוט באמצעי ההגנה והמניעה אשר הוזכרו לעיל בהקשרים של SQL Injection, Cross-Site Scripting, וגם בהקשרים של איומים נוספים כגון חשיפת מידע רגיש והזדהות שבורה. בנוסף, למקרה של תקיפה המתרחשת על אתר מסויים, מומלץ למפתחי האתר לגבות את המידע הנוגע לו בשרת נתונים ואו אמצעי אחסון חלופי, ובאותה מידה למשתמשי האתר מומלץ לגבות את המידע החשוב להם באחד מהאמצעים שלעיל.

כמו כן מכאן ניתן להסיק כי אמנם ריבוי משתמשי האתרים השונים מגדיל את הסיכון שלהם להיחשף לסכנות, ומנגד ככל שיותר משתמשים ומפתחים של אתרים ויישומים ישתמשו באמצעים אלה כך רמת המודעות לסכנות, האמינות והאבטחה של האתרים והיישומים השונים תעלה בהתאם.

ביבליוגרפיה :

-שיעור שני-מהו אתר אינטרנט

אתר אינטרנט – ויקיפדיה

שרת – ויקיפדיה

דפדפן – ויקיפדיה

URL – ויקיפדיה

OWASP Top 10 - 2017

ניווט באתר אינטרנט – ויקיפדיה

FullStack-B-2020/URL.png at master · rayahazi/FullStack-B-2020 · GitHub

World Wide Web – ויקיפדיה

סיומת אינטרנט – ויקיפדיה

Introduction-to-Cyber-2020/SQL injection.md at master · rayahazi/Introduction-to-Cyber-2020 · GitHub

Introduction-to-Cyber-2020/XSS attack.pdf at master · rayahazi/Introduction-to-Cyber-2020 · GitHub

CWE - CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') (4.1)

SQL Injection Prevention

DOM Based XSS Software Attack | OWASP Foundation

Cross Site Scripting (XSS) Software Attack | OWASP Foundation

DOM based XSS Prevention

Cross Site Scripting Prevention

Types of XSS | OWASP

אבטחת אתרים | מערך הסייבר הלאומי