פרטי הבודק : יגאל נאמן 300369972

תאריך 26.8.19

שם התרחיש : DOM XSS(Cross Site Scripting), RFI

הסבר תרחיש :

Cross-site scripting או בקיצור XSS היא אחת ההתקפות הנפוצות ביותר כיום ברמת היישום לתקיפת משתמשים, על ידי הסתננות לתוך יישומי אינטרנט.
ההתקפה גורמת ופוגעת בפרטיות הגולשים באתר האינטרנט הנגוע ויכולה להוביל להפרה של פרטיות הגולש, אשר פרטיו נגנבים או מטופלים.
שלא כמו רוב ההתקפות, אשר יש בהן שני צדדים - תוקף ואתר האינטרנט,
התקפת XSS כוללת שלושה צדדים - תוקף, לקוח (קורבן) ואתר האינטרנט הפגיע.
מתקפת XSS היא סוג של הזרקה, שבה סקריפטים זדוניים מוזרקים לתוך אתרי אינטרנט הנחשבים מהימנים. ההתקפה מתרחשת כאשר תוקף מצליח להעלות לאתר המותקף קוד זדוני, בדרך כלל בצורה של סקריפט הפועל בצד הדפדפן, וקוד זה מורד ומורץ בדפדפנים של משתמשי קצה תמימים. הפגמים המאפשרים להתקפה זו להצליח הינם די נפוצים ויכולים להופיע בכל מקום בו יישום אינטרנט משתמש בקלט משתמש ללא אימות או קידוד מתאים, ואינו מזהה כי הועלה לאתר קוד ולא נתונים כקלט.
לדפדפן של הקורבן אין כל דרך לדעת כי הסקריפט המועבר אליו אינו ממקור מהימן, והדפדפן ימשיך בביצוע ויפעיל את הסקריפט.
בהפעלתו, הסקריפט הזדוני יכול לגשת לכל העוגיות (Cookies) הנשמרות בדפדפן, מידע על התקשרויות, או מידע רגיש אחר הנשמר על ידי הדפדפן בשימוש אל מול האתר המותקף ואף לשכתב את תוכנו של דף ה-HTML.
קשה להגן מפני XSS בצורה מלאה וחייבים להיות מודעים להשלכותיהם של ביטויים שונים תוך כתיבת הקוד, להכיר את המתקפה והדרכים בה היא פועלת על מנת לכתוב קוד בטוח שיגן על האתר והגולשים המבקרים בו.
יישומי אינטרנט מודרניים אינם דפי HTML סטטיים.
לרוב הם דינמיים ומלאים בתוכן משתנה.
הם מושכים נתונים ומידע ממקורות רבים ושונים. מידע זה מאוגד עם דף האינטרנט ומוצג לגולש בדפי האתר על ידי הדפדפן, הוא יכול להכיל טקסט פשוט או תמונות והוא יכול להכיל תגי HTML כמו, <p> לסעיף, <img> לתמונה ו< script> לסקריפט.
בהתקפת XSS טיפוסית התוקף מדביק דף אינטרנט לגיטימי על ידי הזרקת הסקריפט הזדוני שלו לשדה קלט שלא הוגדר כראוי. גולש אקראי שיכנס לדף הנגוע, הדפדפן יוריד ויפעיל את הסקריפט המוזרק, ויגרום להתנהגות לא רצויה. לדוגמה, פוסט פשוט על הקיר בפייסבוק עשוי להכיל תסריט זדוני המוזרק לתוכו. במידה ולא יסונן על ידי השרתים של פייסבוק, בעת ביקור של גולש בפרופיל הנגוע , ירד הסקריפט ויופעל בדפדפן של המבקר התמים.
חשוב להדגיש כי בהתקפה מסוג זה התוקף יכול לבצע פעולות בשמו של המשתמש בשירות על ידי ניצול מגבלות בפרוטוקול HTTPואף לגנוב את מזהה המשתמש. (shadow)

פרצות "זמניות."

1. לרוב, את פרצת ה XSSהזמנית אנחנו נראה בכתובות של אתרים ,**כאשר כתובתם דינאמית.**
2. כתובת דינאמית תהיה, נניח – http://mysite.com/index.php&search=hello ,בכתובת זו אנו עושים חיפוש אחר המחרוזת ,"hello" והיא משתמשת בשיטה בשם GET לשליחת הנתונים שלה, לפיה הפרמטרים מועברים דרך הכתובת) לדוגמה search :הוא פרמטר, שמציין לעמוד index.php באיזו מחרוזת להשתמש לצורך החיפוש. במקרה זה, המשתמש החליט לחפש.(hello
3. אופציה למה שיוצג למשתמש לאחר חיפוש שכזה הוא: "חיפשת את המחרוזת ,hello ולהלן התוצאות". אנו רואים שפלט הדף קשור באופן ברור למחרוזת שהזנו, ולכן נשחק מעט עם הכתובת.
4. נשנה את הכתובת ל . <http://mysite.com/index.php&search=<script>alert("hello")</script
מה שהכנסתי בפרמטר החיפוש הוא בעצם סקריפט קצר בג'אווה סקריפט, שמטרתו הוא להקפיץ הודעה למשתמש.
5. נכנס לכתובת, ואם תוצג למשתמש הודעת שגיאה עם הכיתוב ,hello נדע בוודאות שהאתר פגיע.

רבים חושבים "למה זה פגיע, הרי זה מוצג רק אצלנו" – וכאן קבורה הבעיה.

באפשרותנו לשלוח קישור שכזה לאדם אחר המשתמש באותו אתר, כאשר בפרמטר search נוכל להכניס סקריפט שישלוף את העוגיות וישלח אותן ישירות ל Cookie Stealerשלנו. כשהמשתמש ילחץ על הקישור (וזה ממש לא בעייתי לעשות דבר שכזה עם קצת הנדסה חברתית (

- כאשר מתייחסים ל RFI הכוונה שלי שהיא כוללת יכולת לגשת לקבצים מרוחקים **:(RFI- Remote File Inclusion)** תהליך של הכללת קבצים מרוחקים, מתרחשת כאשר דף מקבל כקלט את הנתיב אל הקובץ שיש לכלול, אך הקלט אינו מסונן ומאומת כראוי ומאפשר הזרקת כתובת חיצונית והפעלת קבצים מרוחקים במקום הקלט המקורי.

Document Object Model Based Cross Site Scripting: (DOM BASE XSS)

קיימת לרוב בסקריפט צד לקוח אשר משתמש בקלט כדי לייצר עמוד ,HTMLבלי לוודא כי קלט זה אינו מכיל קוד זדוני.

לדוגמה, קוד JavaScript אשר מקבל כתובת אינטרנט כקלט, ומשתמש בה לייצור עמוד ,HTML בלי לוודא שהכתובת עצמה אינה מכילה קוד, יכיל פרצת XSS מקומית.

בניגוד לשניים הקודמים, שם הפגיעות היא ב (server side code-חוסר ב ,) input validation-כאן הפגיעות היא בקוד הרץ ב.browser

דוגמאות:

תוקף יצר לינק לאפליקציה בנקאית עם שילוב קוד עוין בלינק ושלח זאת באימייל למותקף

המותקף קיבל אימייל, מקליק על הלינק, והמשתמש שולח את הסקריפט טמון בבקשה אל אתר הבנק

הוא לא מזהה זאת ולכן שולח response עם הסקריפט והוא יוצא לפועל ע"י הדפדפן של המותקף .

הסקריפט הרץ על דפדפן המותקף שולח את ה -cookie-וה info session אל התוקף

מצורפות תמונות להמחשה שמציגה בפנינו את הפגיעות:

Advisory | Request1 | Response1 | Request2 | Response2

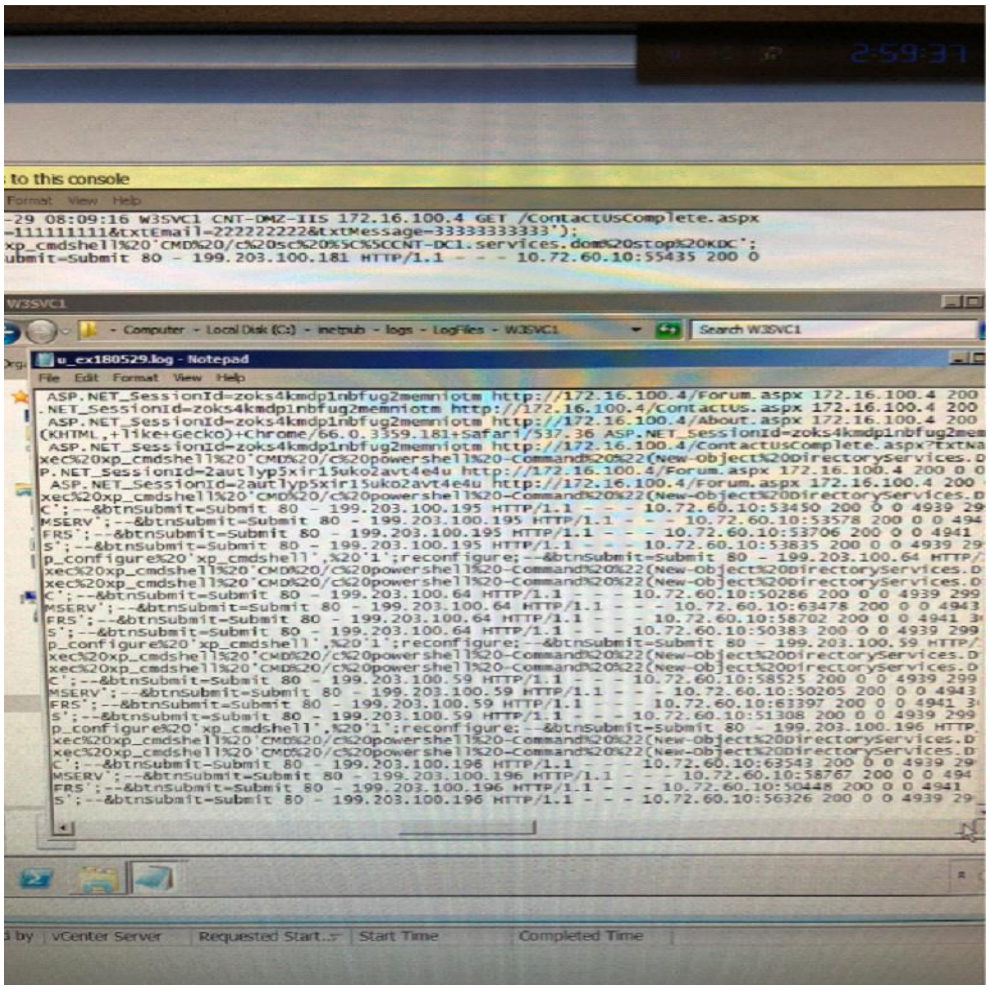
JavaScript injection (DOM-based)

Issue: **JavaScript injection (DOM-based)**
Severity: **High**
Confidence: **Firm**
Host: **http://172.16.67.136**
Path: **/WebGoat/attack**

Issue detail

The application may be vulnerable to DOM-based JavaScript injection. Data is read from **document.URL** and written to **eval()** via the following statements:

- ur=document.URL;
- pr=ur.substring(x+1,ur.length).split("&");
- nv=pr[i].split("=");
- mn="menu"+unescape(nv[1]);
- eval("trigMenuMagic1("+mn+"","+opt+")");



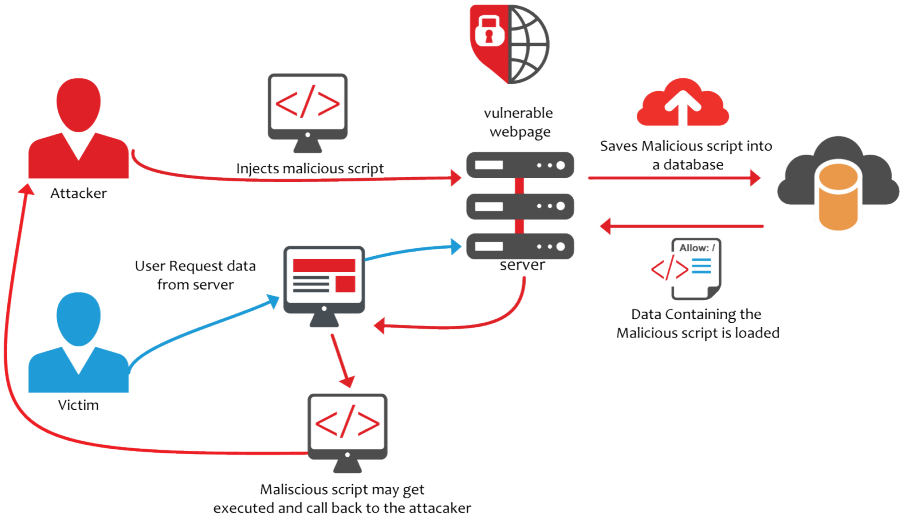
בתמונה השנייה כאן למעלה ניתן להבחין בבקשות GET שהגיעו ממחשב התוקף למקור היעד האתר הנתקף (שמוגן על ידנו) במערכת הבדיקה שלנו שזיהתה את ניסיון ההתקפה שלא צלח כיוון שהמערכת DMZ -CNT שלנו שמנוטרת 7\24 והצוות שלנו חסמו את הפרצה וטיפלו בבעיה.

מטרות וחשיבות הבדיקה שלנו:

באופן כללי זוהי התקפה עם רמת סיכון **גבוהה**. ובמידה והתקלה 'הפרצה' הזו לא תטופל\תתוקן בהקדם, לתוקף שיש לו פוטנציאל בסבירות גבוהה מאד, וכמעט שבוודאות יוכל\לפרוץ ו\או לממש את הפריצה באתר שלכם ודרכו לגנוב מידע ממסד הנתונים של הארגון, ובנוסף התוקף יוכל גם כן להתקיף אנשים תמימים ברשת בזכות הקוד הזדוני שמושתל בשרת.

להלן דיאגרמה (Design pattern) (לאתר אינטרנט לא מוגן)

שמסבירה לנו כיצד ואיך כל זה בעצם קורה הוצאה לפועל ל המתקפה מבחינת מימוש (תמונה להמחשה):

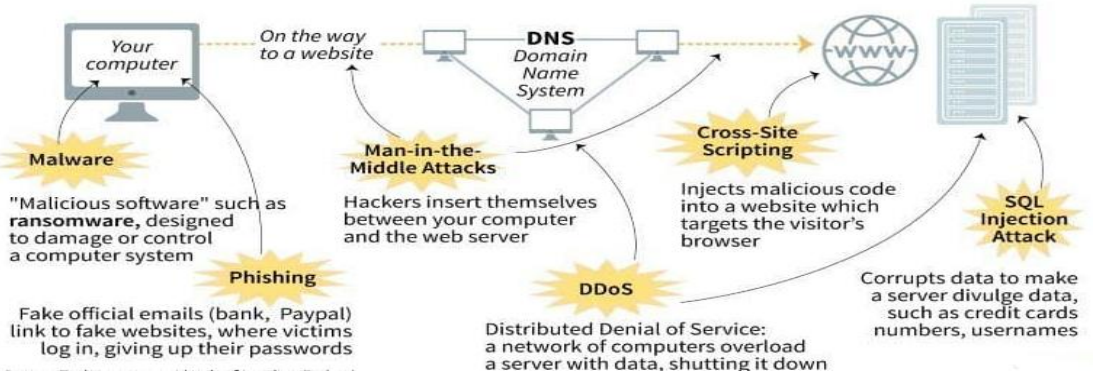


הסבר

כמו שאנחנו יכולים לראות, התוקף ניגש לשרתים 'SERVER, ושותל שם קוד זדוני

לאחר מכן משתמש תמים נכנס לקישור לאתר והקוד הזדוני יורד לו למחשב באופן אוטומטי ללא ידיעת המשתמש ומבצעת EXECUTE(הרצה) מאחרי הקלעים.

לתוקף יש שליטה על ה- DATABASEמסד הנתונים וכמו כן גם על המחשב של מי שנדבק בקובץ הזדוני בדף הרישום.



כמו שניתן לראות לפי התרשים הנ"ל, תוקף בעל פוטנציאל שעלול לממש את פריצת האבטחה \ואו במידה שמימש אותה כי לא הייתם מוגנים, התקוף יכול לממש דרך השרת שלכם התקפות PHISHING,מתקפות מניעות שירותDDOS .

<u>כמו כן התוקף יוכל להזין שאילתות בSQL ולגנוב מידע ממסד הנתונים, כגון:</u>

<u>שם משתמש וסיסמאות</u> <u>פרטי כרטיסי אשראי</u> .

- לסיכום:
  - לאור הנסיבות של בעיה מסוג זו, ההמלצה היא: שיש להתייחס לבעיה מסוג זה <u>ברצינות מאד</u> ולטפל בה <u>בדחיפות</u> ואין לדחותה כיוון שלפריצת האבטחה מסוג זה יש פוטנציאל <u>הרסני גבוהה</u> בין אם זה על הארגון ובין אם זה על משתמשים תמימים ברשת, אי ולכך ובהתאם לזאת ההמלצה היא <u>שלא לדחות</u> טיפול בנושא זה.

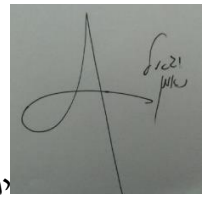
<u>המלצות ליישום</u>: בין אם האתר מאובטח או לא יש להתייחס לשלושה מעגלים:

1. מניעה – כתיבת קוד אתר באופן מאובטח, כאשר כל שדות הקלט עוברים אימות וקידוד.

לדגומא: תהליך מניעה לטווח הרחוק הוא לייעץ לארגון לבצע בקרת קלט כלומר שלא יוכלו להזין ערכים שאינם חוקיים;

2. הגנה – שימוש בכלי הגנה; כגון WAF , IPS , גיבוי האתר והכנת תכנית פעולה להתאוששות.
3. תהליך בדיקה בדרך כלל מתבצעת ע"י - סריקת האתר להערכה וזיהוי פגיעויות, באמצעות שימוש בתוכנות או שירותי ענן אוטומטיים, או היעזרות בבדיקות חוסן של גורמים חיצוניים.

בכבוד רב,



יגאל נאמן.