הצעות לפרויקטים אבטחת מידע – 236349

מנחה: ד"ר גבי נקבלי gnakibly@cs.technion.ac.il

פרויקט מס' 1: חיפוש חריגות HTTP Cookies

עוגיות HTTP מאפשרות לשרתי web לשמור מידע על ה- client. לאחר הבקשה הראשונה לשרת ה- web ה- client מקבל עוגיה מהשרת המכילה מידע שיכול להיות שימושי לשרת בעת עיבוד הבקשות הבאות מה- client. כאשר ה- client שולח בקשות נוספות לשרת, הוא מצרף לכל בקשה את העוגייה. עוגיית HTTP קשורה לשם דומיין מסוים. ה- client ישלח בחזרה עוגייה מסוימת רק עם בקשות המיועדות לשם הדומיין שאליו שייכת העוגייה. למשל עוגיות ששיכות ל- google.com לא יישלחו לבקשות לאתר ynet.co.il.

קיימות אינדיקציות (ראה למשל: http://www.hackerfactor.com/blog/index.php?/archives/650-C-is-for-Cookie.html) שמראות כי לעיתים בקשות HTTP מ- clients נשלחות עם עוגיות שאינן קשורות לדומיין שאליו מיועדת הבקשה. דבר זה עלול להוות סיכון משמעותי לפרטיות של משתמשים שכן עוגיות אלו חושפות מידע על המשתמשים.

מטרת הפרויקט היא לחקור את התופעה של שליחת בקשות HTTP עם עוגיות שאינן קשורות לדומיין הרלוונטי, להבין אילו עוגיות נשלחות כך, למה הן נשלחות, באיזו היקף תופעה זו קורית והאם היא מהווה סיכון לפרטיות של ה- clients ששולחים עוגיות כאלו.

במהלך הפרויקט הסטודנטים יחפשו עוגיות חריגות בתעבורת HTTP ברשת אמיתית. לשם כך הסטודנטים יכתבו סקריפט ב- Python שמאזין לרשת ומחפש עוגיות חריגות, הסקריפט ישמור את העוגיות בבסיס נתונים וינתח את הממצאים.

דרישות קדם: ידע ב- python.

פרויקט מס' 2: תרמיות פרסומות ב- Web (Ad fraud)

פרסומות באינטרנט בד"כ מתבצעות ע"י שידוך בין מפרסם (למשל חברת קוקה-קולה) לבין בעל אתר שמוכן להציג פרסומות באתר שלו. השידוך לרוב נעשה באופן אוטומטי ע"י רשת פרסום כדוגמת Google Ad Network. המפרסם משלם לרשת הפרסום שמעבירה את הכסף (לאחר ניכוי עמלה) לבעל האתר. התשלום מבוצע בד"כ על פי מספר הצפיות של הפרסומת באתר (לעיתים התשלום מבוצע גם על פי על פי מספר ההקלקות של הפרסומת באתר).

מודל תשלום זה פרוץ לתרמית נפוצה שבה בעל האתר מייצר תעבורת דמה לאתר שלו ע"י תוכנה שמדמה גלישה של משתמשים. גלישות הדמה הללו גורמות לכך שפרסמות "נצפות" באתר ועל כן בעל האתר מקבל כסף על כך. הנפגע העיקרי מתרמית זו הינו המפרסם שמשלם על צפיות מדומות, ובמידה

פחותה גם רשת הפרסום נפגעת עקב פגיעה פוטנציאלית במוניטין למרות שהתרמית לא עולה לה כסף באופן ישיר.

רשתות פרסום מנסות להילחם במידה מסוימת בתרמיות מעין אלו, אך עדיין הן קיימות באופן לא זניח באינטרנט. בפרויקט זה ננסה למצוא שיטה שבה המפרסם עצמו יכול לגלות האם הפרסומת שהוא מציג באתר מסוים אכן נצפית ע"י משתמש אמיתי או לא. השיטה תנצל את העובדה כי פרסומת המוצגת באתר כוללת בהרבה מקרים גם קוד Javascript שנתון לשליטתו הבלעדית של המפרסם ובעזרתו הוא יכול לגלות את התרמית.

בפרויקט ננצל את העובדה כי במרבית המקרים כאשר תוקף מייצר גלישות דמה מרובות לאתר מסוים, גלישות אלו אינן מוצגות בפועל על מסך (לרוב התוכנות של התוקף רצות בחוות שרתים שכלל לא מחובר להם מסך). הצגת האתר על מסך תגרום להאטה משמעותית של קצב ייצור הגלישות המדומות. על כן מטרת הפרויקט היא למצוא שיטה באמצעות Javascript שבה ניתן לדעת האם הפרסומת מוצגת על מסך. על השיטה להיות רובאסטית להתחמקויות (למשל שימוש פשוט בתכונה hidden של אלמנט ב-html לא יעזור שכן התוקף יכול לשנות תכונה זו מראש). כיוון אפשרי: מדידות זמנים של פעולות שיכולות לקחת זמן שונה במקרה והן מוצגות על המסך או לא.

דרישות קדם: ידע ב- Javascript

פרויקט מס' 3:

Finding unknown routing protocol deviations in routers

Routing protocols such as OSPF and BGP have open standards (published in RFCs), however commercial routers (e.g. Juniper and Cisco) that implement those protocols may deviate from those standards (deliberately or inadvertently). Such deviation are important to identify from a security point of view as they may be a pose a security vulenrability. We would like to discover those deviations. The problem is that commercial routers are closed-source so it is difficult to directly analyze the router's implementation.

In this project we will use a tool written by Dr. Adi Sosnovich that finds deviations using black-box testing of a router. The tests are generated based on a reference model of the protocol's standard. The tests are then executed against the router and test outputs are compared against the expected output. The tool does this entire process automatically.

The tool was previously used to find deviations of the OSPF implementation in a Cisco router (and was able to uncover several of them). We will extend and adapt this tool so it will be able to find deviations in a **BGP** implementation of a **Juniper** router. The goal of the project is to discover unknown vulnerabilities and report them to Juniper.

Programing languages: C, Python

Required Background**:** Computer Networks, Formal verification (preferably).

Supervisors**:** Dr. Adi Sosnovich, Dr. Gabi Nakibly