מיני פרויקט – נושאים באבטחת רשתות

השגת שם משתמש, סיסמא ותעודת זהות של סטודנטים

מגישים: אוהד דביר ועדן חייט.

מטרת הפרויקט:

מטרת הפרויקט היא לאסוף מהסטודנטים את פרטי שם המשתמש, סיסמא ותעודת זהות שלהם. כל זאת ללא ידיעתם וללא העלאת חשד. הדבר נעשה על ידי שליחת קישור לאירוע מסוים של האוניברסיטה (כל קישור יכול לשמש כאן לעזר שכן לא נעשה כל שימוש בתוכן הדף אינטרנט עצמו). במקום האתר (ההזמנה לאירוע) עצמו, יוצג מסך התחברות לשם מימוש ההזמנה, בו מתבקש הסטודנט להזין את פרטי המשתמש והסיסמא שלו ל-moodle, ולמעשה בין אם יקיש confirm או cancel הוא יועבר לדף האתר המקורי כך שהדבר אמור להיראות לו טבעי ולא לעורר חשד או הרהור משני.

במקביל, התוכנה תתחבר לאתר ה-moodle של הסטודנט ותדלה משם את מספר תעודת הזהות למאגר. על ידי החזקה של 3 גורמי הבטיחות הנ״ל, אנו יכולים לגשת לאזור אישי, המייל ולמעשה לכל גורם של האוניברסיטה בשם הסטודנט ולעשות כרצוננו.

כעת, יהיה ניתן לבצע אוטומציה של תהליכים שונים, על ידי הכלים שלמדנו בפרויקט זה, בקלות יתרה. בין תהליכים אלו – כלולים: ביטול רישום לקורסים, בקשת החזר תשלום לימודים (אפילו עדכון פרטי חשבון בנק לזיכוי), פרטי מגורים וכו׳.

מבנה הפרויקט:

<u>חלק ראשון – שם משתמש וסיסמא:</u>

חלק זה הוא למעשה אתר אינטרנט אותו כרגע אנחנו מריצים על localhost port 8000. האתר מציג דף של מסיבה אשר מטושטש מאחורי בקשה להזדהות באמצעות פרטי האוניברסיטה. כחלק ממבנה האתר נוצרות קריאות POST/GET (פירוט בהמשך) שבהן מועבר המידע של הסטודנט ל-server ולמעשה בכל בחירה שיעשה הוא יועבר לאתר המקורי של האירוע.

<u>חלק שני – תעודת זהות ואימות פרטי משתמש:</u>

חלק זה מתבצע מאחורי הקלעים ואינו תלוי בפעולות המשתמש. את שלב זה מימשנו בדרכים שונות. עיקר ההבדל בין הדרכים נטוע בסוג הפעולה שאותה בחרנו לממש בפרוטוקולים של דף ה-html. מעבר לכך, כל שיטה חשפה בפנינו קשיים וחולשות ולכן המעבר בין השיטות מכיל בין היתר העלאת בטיחות, אפקטיביות ויעילות. עוד יפורט על כך בהמשך.

<u>דרך 1: שימוש בסלניום (GET).</u>

הרעיון המקורי שלנו היה להתחבר בעזרת פרטי המשתמש האוניברסיטאיים לתיבת המייל האוניברסיטאית ומשם לדלות את פרטי תעודת הזהות במעבר כזה או אחר על ההודעות הנשלחות. אולם, התגלה בפנינו ששימוש בתוכנה "רובוטית" על תיבת המייל מחייב אישור מראש של המשתמש בתיבה. אישור שכזה אינו נפוץ ולכן ההתחברות לתיבה בצורה אוטומטית התגלה כלא פרקטי במיוחד.

בשלב זה בחנו אפשרות של עקיפת מנגנון הבטיחות של Gmail והתגלתה לנו ספריית selenium אשר מאפשרת לנו ליצור בוט אשר מדמה שימוש אנושי בדפדפן. בנוסף, בשלב זה גילינו שמשום מה בפרטי המשתמש ב moodle מוחזק מספר תעודת הזהות בתיבה שאינה ברת שינוי. הדבר הפתיע אותנו (שכן אין צורך במספר תעודת הזהות כלל באתר וכן הרעיון שכל הפרטים נמצאים באותה פלטפורמה מעמיד אותה כמטרה פוטנציאלית לרעיונות מהסוג של פרויקט זה). לאור כל זאת, החלטנו לשנות את האסטרטגיה שלנו ולהפנות את השימוש בסלניום למודל, תוכנה שנדמתה בפנינו כחשופה יותר מאשר ה-Gmail.

ספריית selenium הצריכה מאתנו ללמוד את מבנה דפי ה-html של אתר ה-moodle ולאחר מכן להדריך צעד אחר צעד את בחירת התוכנה בכדי להגיע לשדה מספר תעודת הזהות.

יתרונות	חסרונות
השימוש בשיטה קל מאוד ולא מצריך ידע בפרוטוקולי רשת.	איטי – מחייב עבור כל משתמש פתיחת חלון, טעינת דף אינטרנט מרובים.
פתרון מאוד אינטואיטיבי ולא מתחכם כלל.	חוסר ודאות – טעינת הדפים המרובים גוררת תופעות לוואי של שגיאות או המתנה מרובה וביטול התהליך. דבר שמותיר את המהלך לא פתור ולמעשה מחייב חזרה על הניסיונות הללו והיכולת להפריד בין שגיאות שוא שכאלו לשגיאות של פרטי משתמש קונקרטיות.
כתוב כולו ב-Python אין צורך בשפות נוספות.	ודאות 2- ההסתמכות על מבנה האתר ברמה הבסיסית ביותר גורר שכל שינוי בו, ולו הקטן ביותר מבחינת עיצוב אפילו, יכול להוביל לשגיאות כוללות וצורך בעדכון כל השיטה כולה.

<u>שימוש בקריאות GET:</u>

כחלק מהמימוש הראשוני של דף האינטרנט אותו בנינו, עשינו שרשור של פרטי המשתמש והסיסמא בכתובת האינטרנט אותה ה-user מבקש מה-server. כלומר, באם מישהו יאזין לרשת במהלך המעשה, הוא יזהה את הפרטיים החסויים הנ"ל משורשרים לכתובת הבקשה וכך אנו מסכנים את עצמנו ואת האתר שלנו בחשיפה לזדוניות. מעבר לכך, הדבר מסכן את הבלעדיות שלנו על הפרטים הנ"ל שכן הם נהיים חשופים לכל ולא רק לנו.

login?name=ohad&subject=dvir	301

זהו צילום מסך של החבילות הנשלחות כאשר בשדה ה-username הוקלד "ohad" ובסיסמא "dvir".
ניתן לראות את שרשור הפרטים הנ"ל בכתובת אותה מבקש המשתמש לקבל בחזרה מה-server.

דרך 2: שימוש ב-BeatifulSoup:

שיטה זו עדיין משתמשת בקריאות ה-GET אולם כעת במקום להשתמש ב-selenium על מנת להשיג את פרטי תעודת הזהות אנחנו משתמשים ב-serialization של דף ה-html. הדבר היה קצת מורכב שכן הוא חייב אותנו למצוא ולשמור את ה-token הייחודי שנוצר לכל משתמש בשעת חיבורו ל-moodle אולם לאחר ששימרנו אותו, יכולנו ליצור קריאה ישירה מהתוכנה שלנו (ולא מדפדפן) לאתר המודל בבקשת התחברות "אותנטית".

על ידי קריאה זו, שימרנו אצלנו את דף ה-html עצמו וחיפשנו רק את שדה תעודת הזהות לשם שליפתו לשימוש שלנו. למעשה, בדרך זו, כל הפעולות נעשות ישירות בשעת ההזנה של המשתמש. אנחנו בודקים באותו תהליך אם הפרטים שניתנו על ידי המשתמש נכונים ואם כן מכניסים למאגר ישירות את ה3 הפרטים. אחרת אנחנו לא טורחים בכלל והפרטים השגויים אינם מצטברים אצלנו לשווא. מעבר לכך, שיטה זו מוגנת יותר מפני שגיאות שכן אנחנו לא משתמשים במבנה דף ה-html, לפחות לא ברמה הכי בסיסית ולכן רק שינויים מורכבים במבנהו יאלצו אותנו לשנות את הקוד שלנו.

יתרונות	חסרונות
המערכת פועלת באופן אוטומטי ולא מצריכה 2 שלבים כמו בשימוש ב-selenium.	השימוש בקריאות ה-GET עדיין חושף אותנו ומסכן את הרעיון שלנו.
אין שמירת פרטים מיותרים (שגויים) במערכת ואין צורך לוודא שאין. באם לא הושגה תעודת זהות הפרטים כלל לא נשמרים.	ידע קצת יותר מעמיק במבנה דפי html והבנה של מבנה הקריאות וה-header של ההתקשרות לשם מציאת, יצירת, שימור ושליחה מחדש של ה-token.
מהירות – אנחנו לא צריכים להפעיל דפדפן ולטעון דפי אינטרנט כלל. מעבר לכך, הטיפול שלנו בשגיאת התחברות למודל מהיר ואינו מעכב אותנו כלל.	

דרך 3: שימוש ב-BeatifulSoup תוך שימוש בקריאות POST:

כעת לאחר שמהירות, יעילות וודאות השיטה שלנו עלתה פנינו לתקן את בעיית הבטיחות שבה. השימוש בקריאות POST מאפשר לנו לשלוח מידע מדף ה-html אותו יצרנו לסרבר שמריץ את התוכנה באופן חסוי כך שהאזנה, לפחות פשוטה, לא תחשוף את מהות האתר ואת המידע שהוא מעביר לסרבר בו הוא רץ. שינוי שיטת הפענוח של דף ה-html של ה-Session לצורה התואמת אפשרה לנו בקלות יתרה לעשות בדיוק כפי הדרך הקודמת רק בבטיחות טובה יותר.

login	301
178483?seller_code=jiclKp8cScf	200

צילום מסך של אותן חבילות הנשלחות מהאתר ל-server המריץ אותו. כעת ניתן לראות כי המידע הנשלח חסוי ואינו גלוי כמקודם.

מה למדנו מהפרויקט:

ראשית הכרנו את בסיסי השפות html, JavaScript, CSS. כולם היו חלק מתהליך הלימוד שלנו והמימוש של הפרויקט גם אם בחלקים צדדיים. כמו כן, למדנו לעבוד עם ה-Developer Tools ולגשת לחבילות רשת שונות ולעקוב אחרי התנועה הרשתית של המחשב. לבסוף חשוב לציין כי למדנו כיצד לפענח את מבנה דף ה-html ברמה עמוקה ועל ידי כך להשתמש בספריית selenium.

עיקר העבודה שלנו היה על אופן ההתקשרות של user עם ה-sever. כיצד חבילות עוברות, מהם headers, מהו token, קריאות POST/GET והיתרונות והחסרונות של כל אחד מהם. למדנו כיצד לבצע Serialization לקובצי html ולהתנהל איתם באופן ישיר ולא רק דרך תוכנות עזר כמו selenium.