

פרויקט באבטחת מידע

דו"ח סיום

חורף 2022-2023

Resilient C&C Communication Based on Public Infrastructure

מנחה: עמיחי שולמן

מגישים: אנה נובין וברק סופיר

רקע

שרתי שליטה ובקרה מהווים את אחד המרכיבים המרכזיים בהתקפות סייבר. נכון להיום, רוב תשתיות שליטה ובקרה מתבססות על שרתים מיוחדים ונקודות קצה המופעלים על ידי התוקף, או שרתים כללים שנחטפו על ידי התוקף והושמשו למטרה זדונית. שימוש בשרתים אלה בתצורתם הנוכחית, דורש שילוב של טכניקות מורכבות ומתוחכמות לשם הסתרת והתממת התקשורת בין השרת למחשבים הנתקפים. יתרה מזאת, הקמה והחזקה של שרתי שליטה ובקרה בתצורה זו הינה בעלות גבוהה יחסית, וחסידת אחד השרתים מובילה לעיתים להפלת התשתית כולה. לכן, הקמה של תשתית שליטה ובקרה מוצלחת ומתפקדת לאורך זמן, שמורה לרוב לשחקנים משמעותיים. בפרויקט זה, נבחן חלופה לשליטה הקיימת להקמה והחזקה של שרתי שליטה ובקרה. אנו נתמקד בשימוש בפלטפורמות חברתיות בתור תשתית להעברת תקשורת בין התוקף לנתקף. להערכתנו שימוש בפלטפורמות חברתיות מאפשר להתגבר על רוב האתגרים הקיימים בתצורה הנוכחית, וביניהם עלות נמוכה של הקמה והחזקה, יכולת התאוששות מהירה לאחר חסימה או הפלה, וקושי בזיהוי ונתיחות של התקשורת.

מטרת הפרויקט

מטרת הפרויקט הינה להוכיח שקיימת יכולת להקמת תשתית של שליטה ובקרה של רשת בוטים, המבוססת על ממשק ציבורי ידוע.

תשתית זו צריכה לעמוד במאפיינים הבאים:

1. התקשורת בתשתית השליטה והבקרה תהיה קשה לזיהוי וניתוח על ידי יצירת חתימה או תבנית קבועה שתתפס על ידי מערכות אבטחה כגון IPS, IDS וכדומה. בנוסף התקשורת תהיה בעלת אנומליות נמוכה ותסתווה בתקשורת לגיטימית (כפי שמשמש מדבר עם שרת ליגטימי).
2. חסימה או הפלה של נתיב התקשורת תשפיע על מספר מצומצם ככל האפשר של בוטים המתקשרים עם השרת.
3. היכולת תאפשר התאוששות מהירה של התקשורת בין הבוט לשרת הבקרה בעת ניתוק או חסימה.
4. תשתית השליטה והבקרה תהיה בעלות הקמה ואחזקה זולה ככל האפשר.

כלים וסביבת עבודה

סביבת עבודה

הבוט כתוב בשפת python, בחרנו בשפה זו עקב כך שהיא שפה קלה ומהירה לפיתוח וכן קיימים בה מגוון רחב של ספריות שימושיות אשר נפרט עליהן בהמשך. כמו כן, בחרנו להתשמש ב-PyCharm בתור קומפיילר ו-Git על מנת שנוכל לנהל את העבודה בצורה נוחה ומסודרת.

Selenium

זוהי פלטפורה פתוחה אשר מספקת מגוון כלים וספריות המוכוונות לעבודה עם אוטומיזציה לדפדפנים.

Selenium נתמך על ידי מספר רב של שפות ובניהם python. בנוסף הפלטפורמה מאפשרת שימוש בכל הדפדפנים הפופולריים, כך שבוט המשתמש בפלטפורמה זו, יוכל לעבוד על גבי הדפדפן המותקן על מחשב הקורבן.

הבוט משתמש בספריית webdriver אשר נמצאת בתוך ה-Selenium. לספרייה זו קיימות מגוון אפשרויות, החל מהגדרת הדפדפן ועד לעבודה עם אלמנטים שונים בדף האינטרנטי. חלק מההגדרות המיוחדות שספרייה זו מציעה היא האפשרות לעבוד עם דפדפן במצב המוסתר מעיני המשתמש. דבר זה שימושי לפעולת בוטים זדוניים, מכיוון שכך הנתקף לא יכול לראות כי נפתח אצלו דפדפן בכל פעם שהבוט ירצה לתקשר עם שרת שליטה ובקרה. על מנת לעשות זאת נגדיר:

```
parameters = webdriver.ChromeOptions()
parameters.headless = True # display a user interface
```

ספריית ה-webdriver כפי שציינו מתמשקת עם אלמנטים שונים בקוד המקור של הדפדפן. דבר זה מעניק לבוט את האפשרות לנוע בחופשיות באתר, כפי שיכול לעשות משתמש רגיל באופן ידני, וליצור רצף פעולות כך שהבוט יוכל לבצע הזנה של מידע לתוך תיבת טקסט, לחיצה על כפתורים, שליפת נתונים וכדומה.

הפונקציה המאפשרת את כל המתואר לעיל הינה "find_element_by_X", כאשר X הינו משתנה אשר יוחלף בסוג החיפוש בו אנו נשתמש על מנת למצוא את האלמנט.

ניתן למצוא כל אלמנט בקוד המקור של האתר במספר דרכי חיפוש שונות, נפרט על חלק מהדרכים הללו:

1. ID – ישנם אלמנטים אשר מוגדרים עם מזהה יחודי, ובעזרתו ניתן לפנות ישירות לאלמנט זה
2. CSS – עבור אלמנטים אשר יש להם נראות כגון פתור, שדה טקסט וכדומה קיים מזהה בקובץ ה-CSS אשר אחראי על עיצוב העמוד. לכן, הבוט יכול לאתר את האלמנט על פי הזיהוי שלו ב-CSS
3. XPATH – לכל אלמנט בדף האינטרנט יש נתיב ייחודי משלו שלפיו ניתן לאתר. כלומר איפה הוא נמצא בקוד על פי התגיות של העמוד, למשל "/html/body/div/main/div[1]/div/div/form/fieldset[2]/button".

נציין, שבעת מעבר מדף אינטרנט אחד למשנהו ובין לחיצה על כפתור לבין הופעת התגובה ללחיצה בעמוד, עובר זמן משתנה, כלומר לא כל האלמנטים בעמוד נטענים באופן סינכרוני ובטווח זמן מייד. לכן, הבוט לא יוכל לגשת לכל האלמנטים מיד עם טעינת העמוד או לחיצה על כפתור. כדי להתמודד עם בעיה זו, השתמשנו בפונקצייה נוספת, והיא "WebDriverWait". פונקצייה זו מאפשרת לבוט לחכות זמן מוגדר מראש לאלמנט מסוים עד שיופיע בדף האינטרנט. במידה והאלמנט המבוקש הופיע בעמוד, הבוט ימשיך לבצע את רצף הפעולות שהוגדרו לו. אחרת, הבוט ימתין עד לתום זמן ההמתנה שהוגדר לו, ואם האלמנט בכל זאת לא הופיע, תיווצר שגיאה שתוחזר לבוט שתתריע על אי הימצאות האלמנט המבוקש.

ספריות ופונקציות נוספות

במהלך ריצה של הבוט, הוא נתקל בתגובות המכילות פקודה שעל הבוט לבצע.

נכון לרגע כתיבת מסמך קיימות שני סוגי פקודות שהבוט מסוגל לבצע: פקודת הדפסה למסך ופקודות SHELL אשר מחזירות את הפלט למנהל התשתית.

עבור פקודות הדפסה למסך הבוט משתמש בספריית ה- tkinter אשר למעשה אחראית על כל ה-GUI ב-python, ובעזרתה יצרנו חלון GUI קטן עם ההודעה שבאה מכיוון מנהל התשתית.

עבור פקודות ה-SHELL הבוט משתמש בספריית ה-subprocess האחראית על התממשקות עם פקודות מערכת ההפעלה whoami, ipconfig, וכדומה. הבוט יקרא לפונקצייה Popen בספריית ה-subprocess על מנת להריץ את הפקודות שמגיעות ממנהל התשתית, לאחר הרצתן הבוט יחזיר את פלט הפקודה למנהל התשתית באמצעות תגובה כפי שיפורט בהמשך.

בנוסף הבוט משתמש בספריות לתרגום קול למלל על מנת לבצע עקיפה ל-reCAPTCHA. ספריות אלו הן AudioSegment ו-speech_recognition. תחילה הבוט יבצע המרה של קובץ השמע שנשמר על המחשב מפורמט WAV לפורמט MP3 על ידי הספרייה AudioSegment. לאחר מכן הבוט ישתמש בספרייה speech_recognition על מנת לבצע את התמלול, זה מתבצע על ידי כך שהוא משתמש ב-GOOGLE API אשר שולח את קובץ השמע לעיבוד ומחזיר לבוט תמלול של קובץ ה-audio.

סקר ספרות ועבודות קודמות

בחלק זה, נסקור עבודות קודמות הקשורות לנושא הפרויקט.

Now you C(&C), now you don't by Amichai and Stav Shulman¹

הרצאה שניתנה על ידי עמיחי וסטו שולמן במסגרת הכנס BSidesTLV בשנת 2022.

בהרצאה עמיחי וסטו דנו באבולוציה של תשתיות שליטה ובקרה של רשתות בוטים, בחסרונותיהן המרכזיים של תשתיות הנמצאות בשימוש כיום, ודרכי התמודדות איתן.

עיקר ההרצאה היה הצגת גישה חדשה להקמה וניהול של תשתית שליטה ובקרה, המבוססת על תשתית ציבורית, וגישה לתוקפים בכל רמת מיומנות. גישה זו, בשונה מתשתיות שליטה ובקרה הקיימות היום, בעלת יכולת התאוששות מהירה ושחזור התקשורת עם המפעיל לאחר חסימה או הפלה. בנוסף, גישה זו ניתנת ליישום בקלות ובעלות נמוכה ומגבירה באופן דרמטי את ההתמדה של קמפיינים להתקפה.

נוסף להצגת הגישה עצמה, עמיחי וסטו הציגו שתי הוכחות שניתן להקים תשתית שליטה ובקרה המבוססת על עקרונות הגישה שהציגו, אחת על פלטפורמת ה-Spotify ושנייה על פלטפורמת ה-Discord.

How Cybercriminals Can Abuse Chat Platform APIs as C&C Infrastructures²

מאמר זה מתאר מחקר מעמיק של חברת אבטחה Trend Micro שמטרתו לתת מענה על השאלה האם פלטפורמות התכתובות פופולריות בפרט ורשתות חברתיות בכלל, כמו Slack, Discord ו-Telegram, יכולת לשמש בתור תשתית של שליטה ובקרה ברשת בוטים.

במהלך המחקר, חוקרי Trend Micro בחנו מספר פלטפורמות והוכיחו שניתן להשתמש בהן בתור תשתית שליטה ובקרה, יתרה מזאת, הם גילו שחלק מהפלטפורמות כבר נמצאות בשימוש זה. שתי הדוגמאות המרכזיות לשימוש זה הינן Discord ו-Slack. נפרט על אחת מהן בקצרה.

Discord הינה פלטפורמת התכתובות המיועדת ברובה לתחום משחקי המחשב, אך ישנן גם חברות גדולות המשתמשות בה במהלך עבודתן. ל-Discord קיים API הדורש ביצוע רישום לפלטפורמה. לביצוע רישום נדרשת כתובת מייל בלבד ולא נדרש מספר טלפון. דבר זה מקל על הקמת התשתית. החסרון המרכזי של פלטפורמה זו הינו הגבלה על גודל הקבצים שניתן לעלות במסגרת פרסום הודעה (לא קיימת הגבלה על סוג הקובץ) ולכן פלטפורמה זו אינה נוחה לשימוש לתוקף הרוצה לגנוב ולהעביר מידע.

¹ <https://www.youtube.com/watch?v=WPAU8J1LIQs>

² <https://documents.trendmicro.com/assets/wp/wp-how-cybercriminals-can-abuse-chat-platform-apis-as-cnc-infrastructures.pdf>

בנוסף להוכחה של ייתכנות לשימוש ב-Discord בתור תשתית שליטה ובקרה, החוקרים גילו ש-Discord נכון לכתיבת המאמר כבר נמצא בשימוש זה – ובין היתר משמש לקמפיין של Bitcoin Mining. חשבו לציין, שהחוקרים דיווחו על מקרים אלה ל-Discord וקיבלו מענה מהיר והרבה מהפעילות הזדונית הוסרה.

Turla Malware Obtains C&C Address From Instagram Comments³

כתבה זו מתארת דוגמה שבה קבוצה בשם Turla, המקושרת למודיעין הרוסי, השתמשה באחת ההתקפות שלה ב-Instagram כדי שהנוזקה תקבל את הכתובת של שרת שליטה ובקרה שלה. הקבוצה ביצעה זאת באמצעות פרסום תגובה לאחד מהתמונות שהעלתה בריטני ספיר באינסטגרם, כאשר באמצעות הרצת Regular Expression על תוכן התגובה, שנראה תמים, מתגלה כתובת url המפנה לשרת שליטה ובקרה.

Malware command and control over social media: Towards the server-less infrastructure⁴

מאמר משנת 2020 הסוקר מגמות חדשות בהקשר של שרתי שליטה ובקרה המשתמשים בפלטפורמות חברתיות ושירותי ענן ציבוריים. בפרט, החוקרים זיהו חמש מגמות מרכזיות:

1. הסתרת תקשורת בתוך הודעות ברשתות חברתיות
2. הסתרת תקשורת בתמונות וטקסט תוך שימוש בסטנוגרפיה
3. שימוש באחסון ענן ציבורי להחלפת מידע והעלאת קבצים שנגנבו ממחשבי הנתקפים
4. שימוש באלגוריתמי Domain Generation
5. העברת הודעות של C&C באמצעות תגובות ברשתות חברתיות, ובפרט באמצעות תגובות לפוסטים של אנשי ציבור

חוקרים זיהו מגמות אלה בפלטפורמות כגון YouTube, Instagram, Twitter ועוד.

נוסף על כך, המאמר דן בדרכים שונות להתמודדות עם מגמות אלה, ומדגיש שהתמודדות איתם אינה פשוטה וצריכה לבוא בעיקר מהרשתות החברתיות עצמן, תוך תמיכה מגופי הממשל.

³ <https://www.securityweek.com/turla-malware-obtains-cc-address-instagram-comments/>

⁴

https://www.researchgate.net/publication/347285814_Malware_command_and_control_over_social_media_Towards_the_server-less_infrastructure

The rise of TeleBots: Analyzing disruptive KillDisk attacks⁵

כתבה משנת 2016 שפורסמה בבלוג של welivesecurity של חברת אבטחה ESET המנתחת את פעילות TeleBots. TeleBots הינו שם כולל לערכת כלים דדונית ששימשה בהתקפות סייבר ממוקדות נגד מטרות במגזר הפיננסי של אוקראינה, אך חשוב לציין שיש לה הרבה נקודות דמיון עם קבוצת BlackEnergy שביצעה התקפות נגד מגזר האנרגיה באוקראינה בשנים 2015 ו-2016.

בכתבה ישנה דוגמה לשימוש של Telebot ב-backdoor המתקשר אם התוקפים בכדי לקבל פקודות באמצעות שימוש ב-Telegram Bot API מתוך Telegram Messenger. Telegram Bot API מתבסס על HTTP כך שהתקשורת בין המחשב המותקף לבין התוקפים תיראה כתקשורת HTTP לגיטימית עם שרת Telegram לגיטימי. שימוש זה בנוסף, מאפשר לתוקפים עצמם לשלוט במחשב הנתקף בקלות באמצעות כל מכשיר שמותקן עליו Telegram Messenger, אפילו ממכשיר נייד.

Chat App Discord Abused to Attack ROBLOX Players⁶

כתבה שפורסמה בשנת 2017 בבלוג של Trend Micro המתארת שימוש דדוני בפלטפורמת תקשורת Discord יחד עם ROBLOX, משחק פופולרי מרובה משתמשים עם יותר מ-178 מיליון משתמשים רשומים. המשחק מסתמך במידה רבה על תוכן שנוצר על ידי המשתמשים.

השימוש הדדוני ב-Discord ושילובו עם ROBLOX בא לידי ביטוי באמצעות שימוש התוקפים ב-Discord API המאפשר הרצת קוד שנוצר ע"י המשתמש, ופרט בשימוש ב-webhooks, המאפשרים לתוכנת ה-chat לשלוח הודעה לערוץ ספציפי במידה ותנאי שנקבע מתקיים.

בדרך זו, התוקפים יכולים לגנוב עוגיות של ROBLOX log in credentials ממחשבי הקורבנות שמתקן עליהם Discord, ולשלוח אותן תוך שימוש ב-webhook לערוץ ספציפי ב-Discord. ה-credentials לאחר מכן משמשים להתחברות לחשבון ה-ROBLOX וגניבת ה-ROBUX, כסף וירטואלי המאפשר לקנות דברים במשחק, והחלפתו לכסף אמיתי.

⁵ <https://www.welivesecurity.com/2016/12/13/rise-telebots-analyzing-disruptive-killdisk-attacks/>

⁶ https://www.trendmicro.com/en_us/research/17/h/chat-app-discord-abused-cybercriminals-attack-roblox-players.html

מהלך הפרויקט

מחקר ובחירת פלטפורמה

בשלב הראשון, סקרנו ממשקים ציבוריים ידועים, וביניהם Reddit, במטרה למצוא את הפלטפורמה האופטימלית. חיפשנו אחר פלטפורמה אשר תאפשר הקמת תשתית תקשורת שתהיה קשה לזיהוי וניתוח, קשה לחסימה או הפלה, תאפשר התאוששות מהירה ותהיה בעלת הקמה ואחזקה זולים ככל הניתן.

נפרט על חלק מהממשקים שסקרנו-

- Discord הינו יישום חינמי, עם יותר מ-150 מיליון משתמשים רשומים, המיועד בעיקר לקהילת משחקי המחשב. Discord נותן למשתמשים לתקשר אם משתמשים אחרים באמצעות התכתבויות, שיחות וידאו ושמע, בין אם באופן פרטי או באופן ציבורי (שרתים). השימוש העיקרי הינו בשרתים, אשר הם סוג של קבוצות שמחולקות לנושאים שונים וקהילות שיכולות להיות קטנות או גדולות מאוד. קיימת האפשרות ליצור קבוצה של עד 10 משתמשים, לא כשרת. Discord דורשת כתובת דוא"ל לביצוע רישום ואינה דורשת מספר טלפון. אחד החסרונות המרכזיים של הפלטפורמה הינו שלא ניתן לראות את ההתכתבויות בשרתים ללא ביצוע רישום. אמנם קיימת אפשרות של רישום זמני עם שם משתמש בלבד, אך נדרש לפתור hCaptcha. בנוסף, קיימת הגבלה על גודל הקבצים שניתן לעלות לפלטפורמה. היתרון הינו שניתן לעלות הודעות לשרתים באופן מיידי.
- Spotify הינו שירות הזרמת מוזיקה דיגיטלי עם יותר מ-165 משתמשים רשומים. השירות ניתן באמצעות אפליקציה או ממשק web. לא ניתן לשמוע מוזיקה באמצעות השירות ללא ביצוע רישום, אך ניתן לחפש אחר תוכן באמצעות הממשק ה-web ולצפות בתקצירי הפרקים (אם מדובר ב-Podcasts). החסרון המרכזי של הפלטפורמה שלא ניתן לעלות תוכן באופן מיידי, ולעיתים נדרשים מספר שעות או ימים עד להופעת התוכן ב-Spotify.
- Reddit הינו אתר מדיה חברתית, עם יותר מ-1.5 ביליון משתמשים רשומים, הפועל בתצורה של מערכת לוח מודעות מכוונת. האתר מאפשר למשתמשים הרשומים אליו לשתף תכנים, בכל נושא אפשרי, החל מחדשות ומשחקי וידאו וכלה בסיפורים אישיים. תצוגת התכנים מושפעת מהדירוג שלהם, שמתבצע על ידי המשתמשים, אך משאירה מקום גם לתכנים חדשים שטרם קיבלו דירוג. התכנים מאורגנים לפי תחומי עניין שונים, המכונים subreddits. כאשר כל subreddit מנוהל על ידי מנהל אחד או קבוצה של מנהלים שיכולים לקבוע כללים לפרסום הפוסטים והתגובות בו. Reddit דורשת כתובת דוא"ל לביצוע רישום, ניתן להשתמש גם בכתובת דוא"ל פיקטיבית מכיוון שלא נדרש לבצע אימות, ואינו נדרש מספר טלפון. אחד היתרונות המרכזיים של הפלטפורמה הינו שלא נדרש לבצע רישום בשביל לצפות בתכנים ב-subreddit השונים. יתרון נוסף הינו שניתן לעלות תגובות לאתר באופן מיידי.

לאחר מחקר מעמיק, והשוואת היתרונות והחסרונות של הפלפורמות השונות, בחרנו להקים את התשתית על בסיס Reddit.

הגענו להחלטה זו מהסיבות הבאות:

1. Reddit הינה פלטפורמה מוכרת ופעילה, אשר מכילה subreddit במגוון נושאים מ-memes ועד לחדשות עם כמות גדולה מאוד של הודעות, ולכן ניתן להתסוות בין התגובות
2. ניתן לפתוח חשבון ב-Reddit בחינם עם כתובת מייל פיקטיבית, לדוגמא בפורמט הבא [.{}_@gmail.com](mailto:{}_@gmail.com).
זאת מכיוון שלא נדרש לאמת את הכתובת מתוך חשבון המייל
3. במהלך פתיחת החשבון, המשתמש נדרש לפתור reCAPTCHA, הצלחנו לעקוף זאת, על כך יפורט בהמשך
4. קיימים מגוון רב מאוד של subreddits
5. ניתן לחפש הודעות ותגובות להודעות לפי שם ה-subreddit שבו פורסמה ההודעה או התגובה ומילות מפתח (מילים הנמצאות בהודעות או תגובה מבוקשת)
6. ניתן לפרסם תגובה על רוב ההודעות והתגובות ב-subreddits, למעט subreddits שהוגדרו להם כללים מגבילים ע"י מנהלי הפורום, כגון, כלל המונע מכל מי שאין לו מספיק נקודות karma להגיב על הודעות, אך ככלים על תגובות בדרך כלל נדירים
7. לכל תגובה יש מזהה ייחודי אשר אינו משתנה עם הזמן או בעקבות עדכון התגובה
8. משתמש יכול לעדכן או למחוק תגובה שפרסם ללא הגבלת זמן
9. לפי בדיקה שעשינו, דיווח על תגובה החשודה כבוט אינו מגיע למנהל הפורום, והתגובה אינה מוסרת
10. התגובות מפורסמות ב-Reddit בטווח זמן מידי
11. מדיניות הספאם של Reddit הינה מדיניות שמתמקדת בשמירה על תוכן אוטנטי ומונעת התנהגות של פרסום מידע בכמות גבוהה, מידע שחוזר על עצמו, מידע שאינו קשור לנושא, בוטים שפוגעים או מקריסים את Reddit ונועדו לקדם תוכן, מוצרים או שירותים.⁷

מבנה תשתית התקשורת

בשלב השני, סיכמנו על ארכיטקטורה של תשתית המותאמת לפלטפורמה שבחרנו. היו לפנינו שתי אפשרויות להעברת הודעות משרת שליטה ובקרה לבוט. האחת, באמצעות פרסום הודעות (posts) ב-subreddits שונים והשנייה, באמצעות תגובות (comments) להודעות ב-subreddits שונים.

אמנם גם פרסום ההודעות וגם פרסום התגובות מתבצע באופן מידי, ולכן מאפשר תגובתיות מהירה של הרשת, אך החלטנו לפסול את הדרך הראשונה, ובחרנו לבסס את התשתית שבנינו על תגובות. זאת מכיוון שפרסום הודעה יותר בולט מאשר פרסום תגובה לפוסט. בנוסף, כמו שצינו לעיל, קיימים חוקים והגבלות המוטלים ע"י מנהל ה-

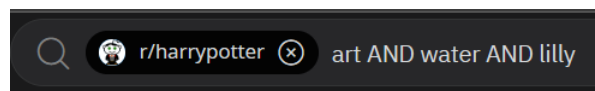
⁷ <https://www.reddithelp.com/hc/en-us/articles/360043504051>

subreddit על התגובות וההודעות המפורסמות בו. לפי בדיקה שערכנו, בדרך כלל יש הרבה יותר הגבלות על הודעות, ומעט אם בכלל לא, הגבלות על תגובות.

בחרנו לפרסם את התגובות ב-subreddit שונים הנבחרים באופן דינאמי שעונים על הדרישות הבאות: מדובר ב-subreddit פופולריים ופעילים עם מגוון רחב של הודעות ותגובות המתפרסמות מדי יום. זאת על מנת להסתוות בין התגובות, ולא למשוך תשומת לב, ובנוסף כדי שתמיד יהיו הודעות חדשות שניתן להגיב עליהם.

מהמחקר המקדים שעשינו, גילינו שניתן למצוא תגובה לפי ה-subreddit שהיא נמצאת בו ומילות מפתח, כלומר מילים שנמצאים בתגובה. לכן החלטנו שהתשתית תהיה בנויה משרשרת של תגובות, כאשר כל תגובה מובילה לתגובה הבאה אחריה באופן הבא: כל תגובה מזוהה ע"י ה-subreddit שבו היא פורסמה ושלוש מילות מפתח המוכלים בטקסט התגובה, וכן כל תגובה מכילה את ה-subreddit ושלושת מילות המפתח של התגובה הבאה בשרשרת.

דוגמה לחיפוש אחר תגובה ב-r/harrypotter subreddit עם מילות מפתח art, water, lilly:



בחרנו להשתמש במילות מפתח ממגוון תחומים, הנמצאות בשימוש יום יומי ואינן מעלות חשד או יוצרות אנומליות – כלומר סיכוי סביר שכל משתמש יחפש מילים אלה במהלך היום יום.

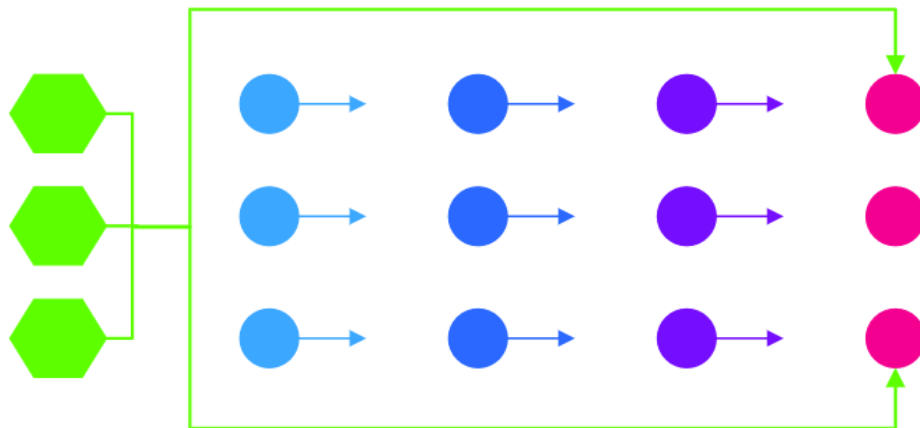
נפרט על מצבים אפשריים של הבוט ביחס לתשתית שליטה ובקרה בהתאם להימצאות ותקינות ה-subreddit ומילות המפתח בתגובה הנוכחית שבה הבוא נמצא, המפנים לתגובה הבאה:

1. בתגובה הנוכחית קיים subreddit ומילות מפתח המפנים לתגובה הבאה, ואכן קיימת תגובה מאומתת (נפרט על תהליך האימות במהשך) העונה על תנאים אלה – מצב תקין שבו הבוט יתקדם לתגובה הבאה
2. בתגובה הנוכחית לא קיים subreddit ומילות מפתח המפנים לתגובה הבאה – מצב תקין המסמן שהבוט הגיע לסוף שרשרת התגובות וטרם פורסמה תגובה עדכנית. הבוט "יידגור" על התגובה הנוכחית עד שהיא תעודכן עם subreddit ומילות מפתח המפנים לתגובה הבאה
3. בתגובה הנוכחית קיים subreddit ומילות מפתח המפנים לתגובה הבאה, אך לא קיימת תגובה מאומתת העונה על תנאים אלה – מצב זה אינו תקין, והבוט והתשתית יפעילו מנגנוני התאוששות וחזרה לתקשורת

תגובה מסוג מיוחד, שממנה מתחילה התקשורת בין הבוט לתשתית שליטה ובקרה, הינה תגובת ה-bootstrap. תגובה זו מהווה גם את אחד ממנגנוני ההתאוששות במצב של תקלה בתשתית. תגובת ה-bootstrap מאופיינת בדומה לשאר התגובות ברשת ב-subreddit שבו היא מפורסמת ובמילות המפתח המובילים אליה. אך בשונה משאר התגובות שהמעבר אליהם מתבצע מתגובה אחרת, ה-subreddit ומילות המפתח של ה-bootstrap נמצאים hardcoded בקוד הבוט. בנוסף, תגובה מסוג bootstrap מתעדכנת באופן תדיר על ידי מנהל הרשת להפנות לתגובה העדכנית ביותר בתשתית. חשוב לציין, כי ניתן לשנות את ה-bootstrap השמור בתוך הבוט על ידי פקודה שתימסר לבוט באמצעות אחת התגובות בתשתית.

מאפיין חשוב של תשתית שליטה ובקרה שבנינו הינו היתירות. בשביל לשמור עליו, פרסמו בכל שלב בתשתית, שניתן לאפיין ע"י ה-subreddit שבו פורסמה התגובה, מספר תגובות זהות, מבחינת מילות מפתח המובילות אליהן, הפקודה שהן מכילות וה-subreddit ומילות המפתח המפנות לתגובה הבאה. מאפיין היתירות נשמר גם בשלב ה-bootstrap.

תרשים של מבנה הרשת:



משושה מסמן את תגובת ה-bootstrap, בהתחלה הוא מפנה לתגובה הראשונה בתשתית, ובהמשך מפנה לתגובה העדכנית ביותר



עיגול מסמן תגובה בתשתית, כאשר צבע העיגול מסמן את השלב בתשתית, המאופיין ע"י ה-subreddit ומילות המפתח, שבו התגובה נמצאת. כל תגובה מפנה לתגובה הבאה אחריה, כאשר התגובה האחרונה



בתשתית, שהיא גם התגובה העדכנית ביותר, אינה מכילה הפנייה לתגובה הבאה, ותעודכן בהתאם בעת פרסום תגובה חדשה

מבנה התגובה


החלטנו על מבנה התגובה הבא:

- טקסט כללי הקשור להודעה שעליה מגיבים
- פקודה שעל הבוט לבצע (אופציונלי)
- שלוש מילות מפתח המזהות את התגובה הנוכחית
- שם של ה-subreddit ושלוש מילות מפתח המזהות את התגובה הבאה בשרשרת
- חתימה

נפרט על חלקים במבנה התגובה שלא פירטנו עליהם קודם לכן:

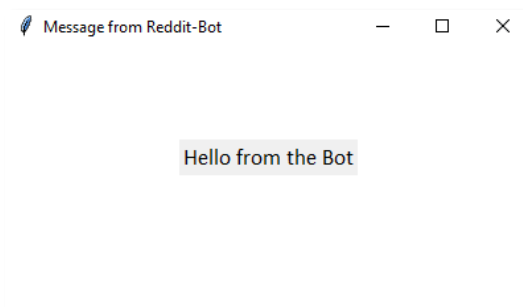
פקודות

יצרנו סט פקודות ייחודי בין הבוט לשרת שליטה ובקרה, בפורמט הבא:

 {command}*{command parameters}

הפקודות שייצרנו הינן הפקודות הבאות:

- הפקודה pm המורה לבוט להדפיס את ההודעה הנמצאת ב-command parameters על המסך
זוהי דוגמא לתוצאת הרצת הפקודה:



- הפקודה rz המורה לבוט לבצע ולהגיב בהתאם לפקודה ב-command parameters כך שפקודות אלה מאופיינות במתן פלט מהקורבן, כגון ipconfig
****צילום מסך של תוצאת הפקודות****

באופן דומה, ניתן להוסיף פקודות נוספות בפורמט זה והלרחיב את הפונקציונליות של הבוט.

חתימה

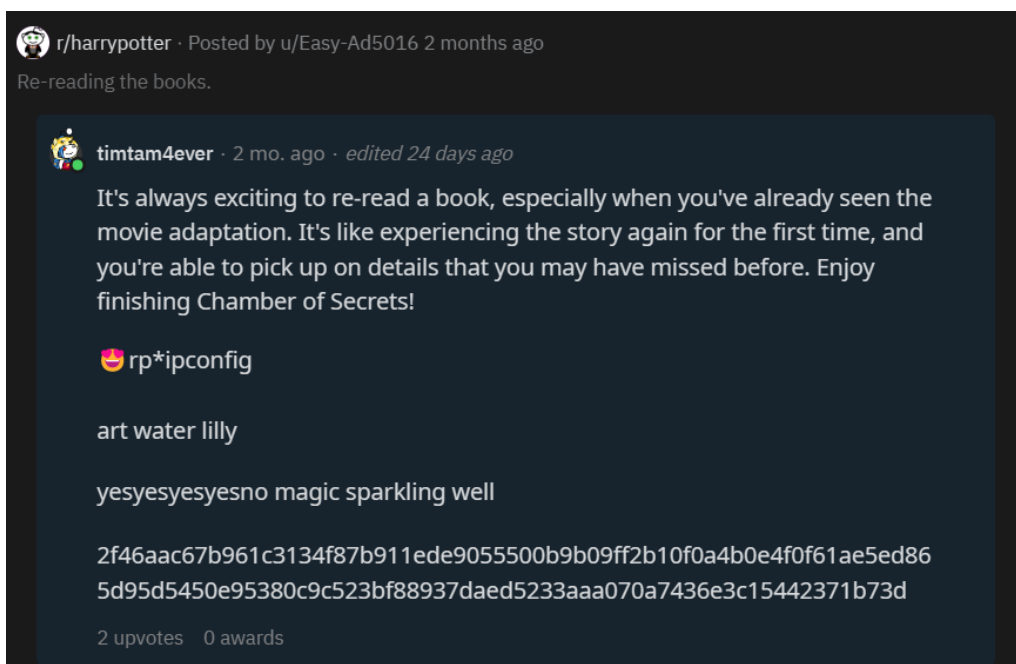
כדי למנוע התחזות לשרת, השתמשנו באלגוריתם ECDSA, המשתמש במפתחות הנגזרים מ-elliptic curve cryptography וייצרנו באמצעותו מפתח פרטי ומפתח פומבי המקושר אליו, כדי לייצר חתימה לכל תגובה על ידי מנהל התשתית ולאפשר לבוטים לאמת את תוכן התגובה בעזרת החתימה.

השתמשנו ב- 256 bit elliptic curve כדי לייצר חתימה באורך 64 byte, שהיא יחסית קצרה מבין אורכי החתימות שבדקנו.

שמרנו את המפתח הפומבי של השרת שליטה ובקרה באופן hardcoded בתוך הבוט. נשים לב, שמכיוון שהבוט מקבל פקודות מהשרת, ניתן להחליף את המפתח הפומבי השמור אצלו.

הצגנו את החתימה בתצורת hex בתגובה, כדי להתאים אותה, נציין שלמרות שרשרת מספרי hex אינה נראת כמו טקסט רגיל, אך אף אחד לא הגיב על כך ב- Reddit ואף תגובה לא הוסרה.

דוגמא לתגובה:



צורת פעולת ה- Controller

מנהל התשתית ינהל את כל הרשת על ידי כלי חיצוני שבנינו אשר נקרא Controller. מטרת ה- Controller הינן ליצור חתימה עם מפתח פרטי ופומבי, תוך שימוש באלגוריתם ECDSA, כפי שפורט לעיל, ולחתום על תגובות שמנהל הרשת רוצה לעלות באמצעות המפתח הפרטי שנוצר.

נכון לכתיבת מסמך זה, יצירת משתמש עבור ה- Controller בפלטפורמת ה- Reddit, בדיקת תקינות התשתית ותיקונה והעלאת תגובות חדשות מתבצעת באופן ידני, אך ניתן בהמשך להוסיף אפשרות לבצע כל זאת באופן אוטומטי.

צורת פעולת הבוט

מבני נתונים ומשתנים מיוחדים

הבוט, כמו כל כלי ותוכנה קיימים, משתמש במבני נתונים על מנת לשמור ערכים ומידע חשוב לשם תיפקודו.

בשלב האתחול הבוט מגדיר ומאתחל לעצמו את המשתנים ואת המבני הנתונים שלו, כך שבעת הריצה של הבוט הוא יוכל לגשת עליהם באופן תדיר ושוטף.

הבוט בנוי על בסיס מחלקה הנקראת Bot. מחלקה זו מקבלת בשלב האתחול שלה את ה bootstrap אשר מכיל את ה subreddit ואת שלושת מילות המפתח של ה bootstrap. כשאר ה-Bot מקבל פרמטר זה הוא שומר אותו בתוך משתנה מחלקה בשם self.bootstrap אשר ערכו יהיה קבוע, אלא אם כן שונה במפורש באמצעות פקודה משרת שליטה ובקרה, לאורך כל ריצת הבוט וזה למעשה יהיה העוגן שלנו למקרה ובו נצטרך להתחיל את כל המסלול מחדש.

בנוסף לכך, בתוך המחלקה של הבוט נמצאים גם משתנים כגון: self.login_email, self.login_password, self.login_username אשר תפקידם לאחסן את פרטי המשתמש שנוצר על ידי הבוט לשם התחברות עתידית ל-Reddit בעת הצורך.

למחלקת ה-Bot קיימים 2 רשימות חשובות (מתוך 4 הקיימות) הנקראות: self.prev_comments, self.rpDoneTasks. התפקיד של self.prev_comments הינו יצירה של שרשרת תגובות בהן הבוט ביקר כך למעשה אנו מענקים לבוט את האפשרות לחזור אחורה במידת הצורך (יפורט בהמשך). התפקיד של self.rpDoneTasks הוא לשמור מזהה עבור כל שלב שבו התרחשה הרצת פקודה ובכך למנוע הרצה כפולה של פקודות. כך למשל אם הבוט הגיע לשלב האחרון ועליו "לדגור" על שלב זה, עד שהשלב יתעדכן עם מילות מפתח הבאות, הוא יבקר בשלב האחרון יותר מפעם אחת ולכן יקרא את תוכן התגובה ובה ידרש להריץ את הפקודה הנתונה למרות שכבר ביצע אותה. כדי להימנע ממצב לא רצוי זה הבוט יבדוק ברשימת self.rpDoneTasks אם הפקודה כבר בוצעה, אם הפקודה זוהתה כבוצעה אזי הבוט יבחר שלא להריץ אותה שוב.

בנוסף למשתני המחלקה של הבוט, קיים משתנה מיוחד אשר מורה לבוט כל כמה זמן לתקשר עם שרת בקרה והשליטה שלו, כך שמשתנה זה יקבע כל כמה זמן הבוט יהיה במצב של שינה ויגרום לבוט לתקשר עם השרת.

במהלך הריצה השוטפת, הבוט עובד עם מנגנון קבלת חייויים עבור המצב שאליו הבוט הגיע, ועבור כל מצב זה קיימות לבוט דרכי התמודדות מיוחדות. נפרט על סוגי החיויים, שקראנו להם status, עבור כל מצב, מתי זה קורה ומה הבוט עושה.

מצבי ה-status:

1. Status == 1 – חייוי המורה על כך שמצאנו תגובה תקינה בעת חיפוש של שלב, כלומר תגובה עם חתימה מאומתת ומילות מפתח שמצביעות לשלב הבא. עם קבלת חווי זה הבוט יחלץ את מילות הפתח הבאות יחד עם ה-subreddit ויריץ חיפוש חדש עם מילים אלו.

2. Status == 2 - חיווי המורה על מצב שבו לא נמצאו מילות מפתח חדשות לשלב הבא (הגענו לסוף השרשרת של התגובות), כלומר תוצאות החיפוש של מילות המפתח הנוכחיות לא הניבו תגובה בה הבוט יוכל להתקדם לשלב הבא, לכן הבוט יבחר על האפשרות של "דגירה" בשלב זה, וזה יעשה על ידי שליחה חוזרת של מילות המפתח הנוכחיות לחיפוש. חשוב לציין כי שלב זה יקרה בלולאה אינסופית עד אשר ימצא תגובה מתאימה שתוציא את הבוט ממצב זה.
3. Status == 3 - חיווי המורה על מצב שבו לא נמצאה בכל השלב הנוכחי שהבוט נמצא בו אף תגובה תקינה, כלומר אין תגובות כלל עם מילות המפתח שמחפש הבוט או שכל התגובות לא תואמות את הפורמט או את החתימה של התגובה, ולכן התשתית תקולה. במצב זה הבוט לא יוכל להתקדם קדימה כי דרכו חסומה ולכן הבוט יבחר לבצע חזרה לאחור על ידי שרשרת התגובות התקינות שהבוט כבר ביקר בהן, עד שהוא ימצא תגובה תקינה. חשוב לציין כי השרשרת תמיד תהיה מעודכנת כך שאם נגיע לשלב תקין בריצה לאחור הוא יצביע לשלב חדש או שלא יהיו בה מילות מפתח, והבוט יעבור ל- status 2.

אתחול

ישנן שתי אפשרויות לאתחול הבוט, האפשרות הראשונה היא שהבוט מורץ בפעם הראשונה על המחשב, והאפשרות השנייה היא שהבוט מורץ לאחר כיבוי המחשב, נפרט על כל אחת מהאפשרויות.

במידה והבוט מורץ בפעם הראשונה על המחשב מתבצעים השלבים הבאים:

אתחול ראשוני

בעת האתחול הראשוני, הבוט מבצע את הצעדים הבאים:

הבוט מבצע אתחול לכל הרשימות השומרות מידע בשביל הפעילות השוטפת של הבוט. לאחר מכן, הבוט שומר את ה- bootstrap שהתקבלת בעת עליית הבוט באופן hardcoded במשתנה וכן שומר את הפתח הפומבי שנמצא hardcoded בקוד הבוט, בקובץ על מחשב הנתקף.

פתיחת משתמש ב- Reddit

לאחר סיום האתחול הראשוני, הבוט מבצע רישום ל-Reddit. הרישום מתבצע ע"י מתן כתובת דוא"ל פיקטיבית בפורמט הבא `{ }@gmail.com` וסיסמא שנבחרה באופן אקראי. בנוסף, Reddit מציע שם משתמש שלא קיים במערכת, שנשלף ע"י הבוט. שלושת הפרטים הללו נשמרים במשתני הבוט ובקובץ config של הבוט, בכדי לא לבצע רישום מחדש בעת אתחול הבוט לאחר כיבוי המחשב. חשוב להדגיש שכפי שציינו, Reddit לא דורש לאמת את כתובת המייל שביצענו רישום באמצעותה, ולכן ניתן להזין כתובת פיקטיבית.

כפי שציינו, קיים מנגון reCAPTCHA שעל הבוט לפתור בכדי להשלים את הרישום. הצלחנו לעקוף מנגון זה, ע"י בחירה באפשרות של השמעת שמע במקום באפשרות של תמונה. גילינו שניתן לשלוף קובץ שמע זה מקוד האתר,

ולשמור אותו במחשב הנתקף. לאחר שמירת הקובץ, השתמשנו בספריית pydub כדי לתמלל את קובץ השמע לטקסט, והזנו את הטקסט שהתקבל כפתרון של ה-reCAPTCHA.

באופן זה, הצלחנו להשלים את תהליך פתיחת המשתמש בהצלחה, נדגיש שלכל בוט נפתח משתמש ייחודי משלו.

במידה והמשתמש נחסם, יש אפשרות להוסיף לבוט יכולת לפתוח משתמש חדש עם אותה כתובת דוא"ל אם כתובת פיקטיבית חדשה, כתלות באופי החסימה.

במידה ובוט מורץ על המחשב לאחר כיבוי והדלקת המחשב מחדש (persistence), מתבצע אתחול שבמהלכו נשלפים כל הנתונים שנוצרו בעת הרצת הבוט בפעם הראשונה מקובץ ה-config הנשמר בזכרון.

פעילות שוטפת

אנו מחלקים את הפעילות השוטפת של הבוט לפי מצב התשתית. כלומר, אנו מחלקים את הפעילות לשלושה מצבים, מצב שבו התשתית תקינה והבוט מתקדם בשלבים השונים בה ע"י מעבר מתגובה לתגובה, מצב שבו הבוט הגיע לסוף התשתית, וטרם פורסמה התגובה הבאה, ומצב שבו התשתית תקולה. נפרט על כל אחד מהמצבים.

מצב ראשון: התשתית תקינה, הבוט לא הגיע לסוף שרשרת התגובות

במצב זה, הבוט נמצא באחד משלבי ביניים בשרשרת התגובות, בין אם עקב ניתוק מהרשת או כיבוי המחשב.

הבוט יחפש, באמצעות שורת החיפוש ב-Reddit, אחר התגובה הבאה לפי מילות המפתח וה-subreddit בתגובה האחרונה שביקר בה. לאחר שימצא תגובות העונות לתנאים אלה, הוא יעבור על התגובות לפי סדר הופעתן, עד שימצא תגובה המכילה חתימה תיקנית. בשלב זה, הבוט ישמור את הקישור המכיל את מזהה התגובה הייחודי במערך ה-`prev_comments`.

במידה והבוט לא ימצא אף תגובה העונה לתנאים אלה, הבוט יעדכן את משתנה ה-`status` ל-3 ויעבור למצב שלישי, המסמל תקלה בתשתית.

לאחר מכן, הבוט יבדוק אם קיימת פקודה לביצוע בגוף התגובה על פי הפורמט שהגדרנו, ויבצע אותה.

אם הפקודה דורשת מהבוט ליצור תקשורת עם השרת באמצעות תגובה על התגובה הנוכחית שהשרת פרסם, הבוט יבצע `log in` ל-Reddit באמצעות שם המשתמש והסיסמא איתם ביצע רישום. אחרי ההתחברות, הבוט יגיב על התגובה הנוכחית, עם הפלט הנדרש בהתאם לפקודה. הפלט מוצפן באמצעות הצפנת קייסר.

חשוב לציין כי במקרה והייתה פקודה לביצוע, הבוט יבצע את הפקודה וישמור את מזהה השלב שבו נמצאת הפקודה ברשימת הפקודות שבוצעו. זאת במטרה למנוע ביצוע חוזר של פקודה שכבר בוצעה, בהנחה שלפי מבנה התשתית, הפקודות בכל התגובות באותו השלב זהות. (להוסיף גם ל-Pm)

לבסוף, הבוט בודק אם קיימות מילות מפתח ו-subreddit המפנה לתגובה הבאה בשרשרת. במידה וכן, הבוט מעדכן את משתנה ה-`status` ל-1 וממשיך בחיפוש אחר התגובה הבאה, כאשר הוא נשאר במצב ראשון.

במידה, ולא קיימות מילות מפתח, אזי הבוט הגיע לסוף השרשרת והוא מעדכן את משתנה ה- status ל-2, וכך עובר למצב השני.

מצב שני: התשתית תקינה, הבוט הגיע לתגובה האחרונה בתשתית

במצב זה, הבוט הגיע לסוף התשתית, כלומר הוא הגיע לתגובה שאין בה מילות מפתח המפנים לשלב הבא.

בפעם הראשונה שבה הבוט יגיע לתגובה זו, הוא יבצע את הפקודה הנמצאת בתגובה, ויסמן שהפקודה בשלב זה בוצעה, כדי לא לבצעה בשנית. לאחר מכן, הבוט "ידגור" על שלב זה בשרשרת, ויבדוק אם אחת התגובות עודכה כל פרק זמן שנקבע.

לפי הגדרת מבנה התשתית, כל התגובות באותו שלב, מעודכנות באופן סינכרוני, וזהות מבחינת הפקודה שיש לבצע ומילות המפתח.

לכן, כל עוד התגובה הראשונה שבוט מצא לא עודכנה, כל השלב טרם עודכן, והבוט ישאר במצב שני, ומשתנה ה- status יהיה 2. במידה והשלב עודכן, הבוט יעדכן את משתנה ה- status ל-1 ויעבור למצב ראשון.

במקרה קיצון, שבו כל השלב האחרון נמחק, הבוט יעדכן את ה- status ל-3 ויעבור למצב השלישי המסמל תקלה בתשתית.

מצב שלישי: התקשורת תקולה, הבוט לא מצליח להתקדם

במצב זה, הבוט לא מוצא אף תגובה המכילה את מילות המפתח הבאות בחיפוש ו/או החתימה אינה מאומתת לאף תגובה בשלב. כלומר, מדובר במצב שבו שלב שלם נמחק, או כל התגובות בשלב מסויים לא מאומתות, מה שיכול לסמן שהתשתית נחקרת ונחסמת או לחלופין, הפוסטים ו/או ה- subreddit בהם הופיעו התגובות נמחקו.

נפריד מצב זה לשני מקרים:

1. מחיקת השלב האחרון בשרשרת - במקרה זה הבוט ינסה לגשת לשלב האחרון, בין אם על ידי "דגירה" על

שלב זה ובין אם על ידי הגעה לשלב זה בפעם הראשונה, אך הוא לא יצליח, עקב התקלה ולכן יעדכן את

משתנה ה- status ל-3. כתוצאה מכך הבוט יחזור בחזרה לשלב שקרא לו, באמצעות מערך ה-

prev_comments.

מצד מנהל התשתית, הוא יצור שלב חדש, כלומר יעלה תגובות תקינות לשלב האחרון, ויעדכן שני שלבים

בשרשרת; את שלב ה- bootstrap כך שהוא יצביע לשלב החדש שנוצר, ואת השלב תקין המאוחר ביותר

בשרשרת, כך שיצביע אל השלב החדש שנוצר.

2. מחיקה של אחד או יותר מבין שלבי הבניינים בשרשרת - במקרה זה הבוט ינסה לגשת לשלבים התקולים אך לא יצליח לעשות זאת ולכן יקבל (על כל ניסיון גישה כושל לשלב) status 3. לכן הבוט יחזור אחורה בשרשרת, באמצעות מערך ה- prev_comments עד לקבלת status 1, משמע מציאת שלב תקין. מצד מנהל התשתית, הוא יעדכן את התגובות בשלב התקין המאוחר ביותר בשרשרת להצביע לשלב החדש שיצור.

מנגנוני התאוששות והתמודדות עם הגנות

נתאר דרכים שונות שבהן חוקרים ינסו להשביח את התשתית שבנינו, נפרט את הנזק שיכול להיגרם לתשתית ונסביר כיצג התשתית תתאושש.

התחזות לשרת

חוקר יכול לפרסם תגובה במבנה דומה למבנה התגובה הלגיטימית. זאת במטרה לנתב את התקשורת לשרת שליטה ובקרה משלו בכדי לחקור את הרשת או לחלופין לבלבל את הבוטים ולהשיג שליטה על הרשת.

המנגנון שבא להתגבר על ההתחזות, הינו מנגנון החתימה. כל תגובה המפורסמת ע"י השרת, חתומה ע"י חתימה של מפתח פרטי ופומבי כפי שפירוטו בצורת הפעולה של ה- Controller. הבוט, בעת חיפוש התגובה לפי מילות המפתח, בודק קודם כל אם החתימה נכונה, על פי מפתח ציבורי השמור בו, ורק במידה והחתימה מאמת הבוט קורא את ההודעה ומבצע את הפקודה בה במידה וקיימת. חשוב לציין, שגם במידה והחוקר משיג את המפתח הציבורי ע"י חקירת הבוט, הוא אינו יכול לחתום איתו על תגובות, כפי שלמדנו בקורס הגנה ברשתות.

נשים לב, שאומנם הבוט צריך לעבור על כל התגובות המכילות את מילות המפתח כדי לאמת את החתימה, וכביכול ניתן לחשוב שהחוקר יכול לעלות מספר רב של הודעות כדי לבצע מעין התקפת DOS על רשת הבוטים, אנו מעריכים שפלטפורמת ה- Reddit תשים לב לפעילות זו ותחסום אותה.

מחיקת תגובות בתשתית

מנהל ה- subreddit רשאי למחוק תגובות על פי ראות עיניו או לחלופין חוקר יכול לבקש ממנו, או מ- Reddit עצמו למחוק תגובות מסויימות.

מצב זה, יכול לפגוע בשלמות התשתית וכביכול לנתק תקשורת בין הבוטים לשרת שליטה ובקרה.

כמו שפירוטנו, הרשת בנויה בצורת שלבים, כאשר בכל שלב, יש מספר תגובות הנמצאות באותו subreddit, עונות על אותן מילות חיפוש ומכילות את אותן מילות המפתח המפנות לשלב הבא, כלומר מתקיים יתרון של יתירות, ומחיקת תגובה בודדת לא תשפיע על שלמות הרשת.

נכון להיום, שלמות הרשת נבדקת בצורה ידנית, אך בהמשך ניתן לבצע זאת באופן אוטונומי.

במידה ונחמקו כל התגובות בשלב מסויים, שלמות הרשת תיפגע, ודרכי התאוששות ממצב זה תלויות בשלב שבו זה התרחש. נפרט על כל אחת מהאפשרויות:

1. מחיקת כל התגובות בשלב באמצע או בסוף השרשרת – במצב זה, בדיקת שלמות הרשת, תתריע על שלב

שאינו קיים, למרות שמילות המפתח אליו קיימות, ולכן בהכרח צריך להיות שלב הבא לפי הגדרת מבנה הרשת. מצב זה יכול לפגוע בבוטים שטרם הגיעו לתגובה העדכנית ביותר, מכיוון שהבוטים ברשת יכולים להימצא בשלבים שונים בתשתית. זאת בעקבות ניתוק זמני מהרשת, או כיבוי המחשב וכדומה. דרך התאוששות ממצב זה, הינה שמירת מזהי כל התגובות שבוט ביקר בהן. הבוט יבדוק כל תגובה, מהעדכנית

ביותר עד ל- bootstrap, והתגובה הראשונה שתהיה תקינה ותוביל אותו לתגובה שטרם ביקר בה, תעצור את ההליכה לאחור, תעדכן את רשימת התגובות בהן ביקר ותמשיך את התקשורת מול השרת. באותו הזמן, ה- Controller יעדכן את התגובה האחרונה לפני השלב שנפגע, להצביע לתגובה העדכנית ביותר. נזכיר, שאין הגבלת זמן על עדכון התגובות. במידה, וכל השלבים נמחקו, הבוט יגש ל- bootstrap ויקבל את המילות מפתח העדכניות באמצעותו.

2. מחיקת כל התגובות בשלב ה- bootstrap – במצב זה, לפי הגדרת מבנה הרשת, הבוט ימשיך לנסות לגשת לתגובות בשלב זה, עד שימצא תגובה תקינה. באותו הזמן, ה- Controller מעדכן שלב זה באופן תדיר, ולכן אנו מניחים שרק לזמן קצר מאוד, לא תהיה אף תגובה בשלב זה.

חתימה על תבניות של תקשורת

במידה וחוקר חקר את הבוט, וגילה את מילות המפתח של ה- bootstrap, הוא יכול לחתום על מילים אלה ולמנוע גישה של המחשב הנגוע לביצוע חיפוש ב- Reddit עם מילות חיפוש אלה.

דבר זה נכון גם לגבי חתימה על מילות החיפוש של אחת התגובות באחד השלבים ברשת.

במצב זה, החוקר יכול לחתום על תבנית זו, ולחסום אותה באופן מערכתי במגוון רב של מחשבים. כתוצאה מזה, הבוטים לא יוכלו לגשת לתשתית התקשורת.

אך, תרחיש זה אינו סביר, מכיוון שלפי הגדרת התשתית אנו משתמשים במילות מפתח אקטואליות הנמצאות בשימוש יום יומי, והתגובות מפורסמות בפלטפורמה פופולרית הנמצאת בשימוש אצל מגוון משתמשים. לכן חסימתם תפגע בחווית המשתמשים ולכן היא דרך אגרסיבית מדי בשביל לחסום את התשתית, ואינה סבירה.

חסימת משתמשים המפרסמים את התגובות בשתית

במידה וחוקר יבקש לחסום את משתמש ה- Controller או את המשתמשים של הבוטים המגיבים על תגובות ה- Controller יכול לכאורה ליהוצר מצב שה- Controller אינו יכול לעלות תגובות או הבוטים אינם יכולים להמשיך לתקשר בתקשורת הדו כיוונית מול השרת.

אך, אנחנו מתמודדים עם מצב זה בשני אופנים. מצד אחד, עלות פתיחת משתמש הינה זניחה וניתן לביצוע בטווח זמן מייד, ולכן במידה והבוט או ה- Controller יזהו כי המשתמש נחסם הם יוכלו בקלות לפתוח משתמש חדש עם כתובת דוא"ל פיקטיבית אחרת או אם אותה כתובת דוא"ל (כתלות באופי החסימה). מצד שני, מכיוון שלכל בוט נפתח משתמש ייחודי משלו, והם אינם מפרסמים הרבה תגובות, מאוד לא סביר שהחוקר יצליח להתחקות ולחסום מאות אם לא אלפי משתמשים (כתלות בגודל הרשת).

חסימת פתיחת משתמש ע"י בוט באמצעות reCAPTCHA

בעת פתיחת משתמש בפלטפורמה Reddit המשתמש מתבקש להוכיח שאינו robot באמצעות מנגנון ה-reCAPTCHA ע"י סימון check או ע"י פתרון ה-reCAPTCHA ע"י זיהוי תמונה.

reCAPTCHA אשר נרכשה על ידי גוגל שומשה החל משנת 2014 לשיטת אימות, אשר כפי שיצויין האימות מתרחש על ידי זיהוי אלמנט מסוים בתמונה או על ידי הזנת טקסט משמיעה. שיטה זו ממומשת עד היום במגוון רחב של אתרים במטרה להפחית פעילות של בוטים.

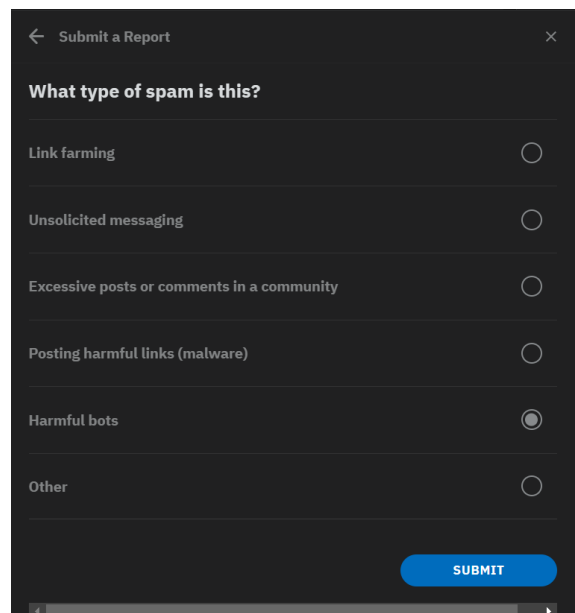
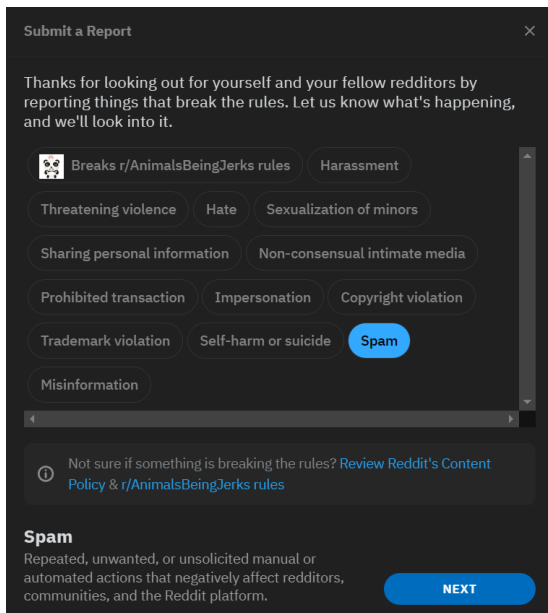
מנגנון הגנה זה, נועד לחסום בוטים מלפתוח חשבון ב-Reddit, ויכל למנוע מאיתנו להצליח להרים ערוץ תקשורת דו כיווני בין הבוט לשרת.

כפי שפירטנו בהסבר על צורת פעולת הבוט, הצלחנו לעקוף מנגנון זה ע"י הורדת קובץ השמע של ה-reCAPTCHA מאתר הפלטפורמה, ותימולולו לטקסט, והזנת התשובה שהתקבלה.

מנגנון דיווח על תוכן זדוני ל-Reddit

ל-Reddit קיים מנגנון דיווח המאפשר למשתמשי הפלטפורמה לדווח על תגובות שלדעתם לא תואמת את מדיניות האתר, או את מדיניות ה-subreddit שבו פורסמה התגובה (מדיניות זו מתווספת על מדיניות Reddit).

לאחר לחיצה על כפתור הדיווח הנמצא ליד כל תגובה, נפתח חלון שבו ניתן לבחור את הקטגוריה שאליה משתייך הדיווח. בכדי לדווח על תגובה החשודה כפורסמה ע"י בוט, יש לבחור בקטגוריית ה-Spam ובהמשך לבחור בקטגורייה של Harmful bot.



לפי מחקר שעשינו, Reddit אינו חוסם בוטים, אלא אם כן מדובר ב-Spam שיוצר עומס על שרתי Reddit או על בוטים שמפרסמים מידע פוגעני.

מכיוון שהתשתית שיצרנו, אינה עונה על קריטריונים אלה, סיכוי סביר שאינה תיחסם ע"י Reddit גם אם תדווח. ואכן מבדיקות שעשינו, ע"י דיווח על תגובות שפירסמו, כ- Harmful bots, מנהלי הפורום לא קיבלו התראה על דיווחים אלה, והתגובות לא הוסרו ע"י Reddit עד לכתיבת מסמך זה.

במצב שבו, בכל זאת התגובות מוסרות, או המשתמש המפרסם אותן נחסם, אנו בקלות יכולים לפתוח משתמש אחר עבור שרת השליטה והבקרה או עבור הבוט, ולעלות תגובות אלה מחדש. זאת מכיוון שאין משמעות ליוזר שפסם את התגובה עבור הבוט, אלא רק על תוכנה ותקינותה. חשוב לציין כי העלות של פתיחת משתמש חדש או העלאת תגובה חדשה היא מאוד זניחה, וניתן לבצע זאת בטווח זמן מידי.

מנגנון עקיפת ה- anti-bot

כפי שהצגנו להעיל, הבוט משתמש בספריית ה- webdriver הנמצאת בתוך Selenium. ספרייה זו פונה לאתר אינטנט בכך שהיא מקבלת שליטה על הדפדפן, בפרויקט זה בחרנו להשתמש בדפדפן ה- Chrome, ושולחת בקשת HTTP או HTTPS (כתלות באתר אליו היא ניגשת) לשרת המיועד.

במצב סטנדרטי בו אנו משתמשים ב- webdriver ניתן לראות בצד הנתקף כי בדפדפן מופיע הודעה האומרת שהדפדפן נשלט על ידי תוכנת צד שלישי.

פרט זה, המצביע על כך שמדובר בבוט השולט על הדפדפן, עבור הלאה גם לשרתי ה- WEB.

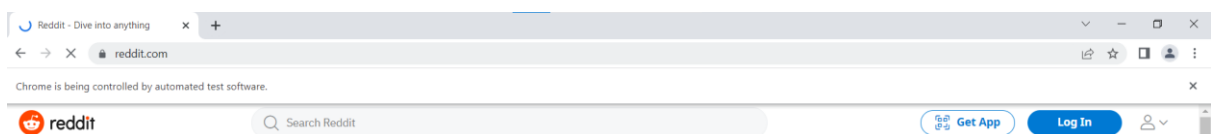
אתרים מסויימים ירצו למנוע מבוטים לבצע בהם פעולות, זאת משלל סיבות שונות, ולשם כך הם ישתמשו במערכות anti-bot שונות המזהות שמדובר בבוט ומגבילות ואף מונעות את פעילותו באתר.

על מנת למנוע את הגבלת פעילות הבוט השתמשנו בספרייה מיוחדת בשם undetected_chromedriver. תפקידה הינו מעקף של מנגנוני ה- anti-bot בכך שהיא מכניסה header רנדומלי של ה- user-agent, דבר העוזר לבוט להתנהג כמו יוזר לגיטימי שגולש בדפדפן רגיל.

בנוסף, ספרייה זו משנה את אופן הלחיצות וההקלדות של הבוט כך שיעשו בצורה כזו שמדמה את אופן השימוש של משתמש לגיטימי. פעילות זו מעניקה לבוט את היכול להסתוות ולהתחמק ממנגנוני הגנה שמנתרים את התנהגות המשתמש.

דבר נוסף שספרייה זו עושה הינו שינוי ב- header המציין כי הדפדפן נשלט על ידי כלי אוטונומי. באופן דיפולטיבי ה- webdriver מכניס ל- header כי יש שימוש בכלי חיזוני, מידע זה עובר בבקשה של HTTP או HTTPS לכיוון השרת – ה- undetected_chromedriver מבצע הסרה של שורה זו.

ניתן לראות כי בתצורה הרגילה מופיע שורה המדווחת כי הדפדפן נשלט על ידי תוכנה אוטומטית:



לאחר שימוש במעקף של ה- anti-bot ניתן לראות כי השורה נעלמת והדפדפן עובד בתצורה רגילה ללא זיהוי כבוט:



לוחות זמנים

לוח זמנים משוער

לוח הזמנים שקבענו בתחילת הפרויקט, המחולק לפי השבועות בסמסטר, הינו:

1. למידה של אפליקציות ושיטות בהן יתבצע שימוש – שבוע 3
2. למידה של תשתיות ציבוריות פוטנציאליות – שבוע 4
3. הקמת פרופיל ברשת הציבורים הנבחרת ויישום התשתית בצד מנהל הרשת – שבוע 5-6
4. כתיבת בוט בעל יכולת תקשורת ויישום התשתית בצד הלקוח – שבוע 6-13
5. ניתוח התוצאות – תקופת מועדי א'
6. כתיבת דוח סופי - תקופת מועדי ב'
7. כתיבת מצגת סיום - תקופת מועדי ב'

לוח זמנים בפועל

לוח הזמנים בפועל היה שונה במקצת מלוח הזמנים המשוער, זאת בעקבות אילוצי לו"ז שונים.

בשבוע הראשון והשני של הסמסטר עסקנו בעיקר במחקר על הפלטפורמות והכלים שאנו רוצים להתשמש בהם בפרוייקט לשם השגת המטרות שהצבנו. בשבוע השלישי התחלנו בפיתוח הבוט, יצרנו את שלד הבוט כך שהוא יוכל לחפש תגובות. בשבוע השישי התחלנו לגבש את כלל הפונקציות, קבענו את מבנה התשתית ויצרנו את התשתית עצמה כדי להוכיח כי התשתית והשיטה שחשבנו עליה אכן עובדים.

בשבוע 10 שכתבנו את הקוד, כך שיעבוד בצורה יעילה יותר עם מחלקות ואובייקטים והתחלנו ליישם את התקשורת הדו כיוונית. החל משבוע 12 ועד לסוף הסמסטר סיימנו את כתיבת הקוד, כולל מימוש הקוד המבצע הרשמה ל-Reddit והעוקף את מנגנון ה-reCAPTCHA.

כתבנו את דוח הסיום לאחר תקופת מועדי ב'.

סיכום

במהלך הפרויקט הוכחנו כי ניתן להקים תשתית שליטה ובקרה בפלטפורמה חברתית Reddit, באמצעות פרסום תגובות לפוסטים ב-subreddit שונים.

התשתית שבנינו בעלת מספר יתרונות בולטים, והם ההתאוששות המהירה אחרי חסימה או הסרה של משתמש או תוכן. זאת בעקבות הטווח הזמן המידי שבו ניתן לפרסם תגובה חדשה ומכיוון שאין משמעות למי פרסם את התגובה בתשתית אלא רק לתוכן התגובה ולאומותה בעזרת המפתח הפומבי של מנהל הרשת, המונע התחזות אליו.

יתרון נוסף של התשתית הינו עלותה הנמוכה, זאת מכיוון ש-Redditt לא גובה תשלום על רישום לאתר או פרסום תגובות. נוסף על כך, הבטים יכולים לחפש ולצפות בתגובות ללא צורך בפתיחת חשבון. מכיוון שהשתמשנו בפלטפורמה פופלרית, חיה ופעילה, השתמשנו במילות מפתח הנמצאות בשימוש יום יומי, והסונו את התקשורת באמצעות תגובות בפורומים בנושאים שונים, החוקרים יתקשו לזהות ולנתח את התשתית, ויתרה מזאת, יתקשו לחסום אותה ללא חסימת מגוון רב של תקשורת לגיטימית.

נוסף על כל היתרונות שציינו, הישג משמעותי נוסף שהצלחנו ליישם בתשתית שבנינו הינו הקמת תקשורת חוזרת מהבוט לשרת שליטה ובקרה. זאת באמצעות רישום הבוט ל-Redditt ופרסום תגובה מטעמו על התגובה משרת שליטה ובקרה המבקשת זאת. הישג זה אפשרי בעקבות מאפייני פלטפורמת Reddit אשר אינה דורשת אימות של כתובת המייל, אימות טלפון ומשתמשת ב-reCAPTCHA. בדומה לתקשורת מהשרת לבוט, גם תקשורת חוזרת מהבוט לשרת הינה קשה לזיהוי וחתימה בעקבות אותן הסיבות.

התשתית שבנינו הינה בעלת מאפיינים ייחודים ומוכיחה שאכן ניתן להקים תשתית שליטה ובקרה מוצלחת בפלטפורמה חברתית. בהמשך, ניתן להוסיף אליה עוד ועוד מאפיינים שיגדילו את מאגר היכולות שלה, וישפרו את חוזקה.