Allot & kamatech project

תיעוד מילולי

מבני הנתונים הנמצאים בשימוש בפרויקט:

* טבלת גיבוב עבור הקונקשנים, התנגשויות מטופלות בשיטת השרשור.
* רשימה מקושרת דו כיוונית – בשימוש לשרשורי הconnections בטבלה, רשימת הטרנזקציות שמכיל כל connection, ורשימה לתאים השימושיים בטבלת ההאש. פרוט בהמשך.
* Connection – מכיל רשימת טרנזקציות, זמן התחלה, זמן סיום, מפתח המורכב מ-3 tuples וגודל.
* Transaction – מכיל זמן התחלה, זמן סיום, מספר פקטות inbound, מספר פקטות outbound, גודל, גודל פקטת inbound מקסימלי ומינימלי, הפרש זמן בין 2 פקטות inbound מקסימלי ומינימלי, הפרש הזמן בין שליחת הבקשה מהקליינט להגעת התגובה הראשונה מהסרבר.
* Five\_tuple – מכיל את השדות server\_ip\_address, client\_ip\_address, client\_port .

מהלך התכנית:

התהליך ניגש לקובץ הpcap וקורא ממנו את הפקטות.

כל פקטה נבדקת, ראשית כל, האם היא שייכת לפרוטוקול udp. אחרת ממשיכים מיד לפקטה הבאה,

וחוסכים הקצאות ופעולות מיותרות.

בשלב הבא נבדוק האם ה-dest port של הפקטה שווה ל443 – הפורט של youtube. אותו בחרתי

להגדיר בdefine בראש הקובץ. או לחילופין האם ה-src port שווה לפורט הנ"ל וכן גודל הפקטה עומד

בטווח שנקבע לפקטות inbound. אם התשובה שלילית, ג"כ נמשיך לאיטרציה הבאה ללא התיחסות

לפקטה.

במקרה שהתשובה חיובית:

ניצור מפתח המורכב מ-3 tuples (אמנם במקור קיימים 5 כאלה, אבל כיוון ששנים מתוכם יהיו משותפים

לכל הפקטות הרלוונטיות שלנו, אין לנו צורך לשמור אותם. השנים הם סוג הפרוטוקול (udp) והפורט

של יוטיוב (443).)

אם הפקטה היא פקטת request, כלומר בגודל שווה או עולה על ה request\_packet\_threshold,

ניגש לטבלת ההאש במקום המתאים ונחפש ברשימה אליה הוא מצביע. (אם אין רשימה כזו ניצור

אותה, ונוסיף מצביע גם ברשימת המקומות המשומשים בהאש). אם מצאנו קונקשן בעל אותם ערכי

Tuples נוסיף את הטרנזקציה החדשה לרשימת הטרנזקציות שלו ונעדכן את הend\_time של

ה connection. אם לא קיים, ניצור אותו ונאתחל לו רשימה בעלת טרנזקציה אחת. נדחוף אותו לראש

הרשימה שבטבלה, ובכך ננצל את עקרון הלוקליות (– סביר להניח שבפרקי זמן צמודים נקבל פקטות

עבור אותו קונקשן, והעובדה שבראשי הרשימות יעמדו הקונקשנים האחרונים שנוצרו מעלה את אחוז

הסיכוי ל"פגיעה" – hit מיד בתחילת החיפוש בהן. כך שגם במקרים בהם ישתרשרו כמות גדולה של

קונקשנים לאותו מקום בטבלת האש, עדין במרבית הפעמים הגישה תהיה ב-O(1) ).

אם הפקטה מיועדת ליוטיוב אך גודלה אינו עונה על הדרוש לrequest, נחפש את הקונקשן המתאים,

ואם נמצא נעדכן את הזמן האחרון שלו ושל הטרנזקציה האחרונה שלו (זו תהיה הראשונה ברשימת

הטרנזקציות, שכן דחיפת טרנזקציה נעשית בראש הרשימה).

לפקטת response:

נחפש את הקונקשן המתאים, במקרה שלא נמצא נתעלם מהפקטה. אם הוא אכן קיים, נבדוק האם

היה צורך לסגור אותו – בשל הפרש הזמן בין הend\_time שלו לזמן הכניסה של הפקטה הנוכחית,

או בשל מספר הטרנזקציות שכבר ארעו בו. אם כן, אם גודל הקונקשן עונה על הנדרש מוידאו הוא

ישלח לפונקציה שתכתוב אותו לקובץ הcsv, ובכל מקרה נמחק אותו מרשימת הקונקשנים אליה היה

שייך. (בשל היותה דו כיוונית, המחיקה תתבצע ב- O(1). )

אם לא, נעדכן את הנתונים בטרנזקציה האחרונה שלו, וכך גם את הsize וה-end\_time של

ה connection.

בסיום המעבר על הפקטות, נעבור על הרשימה בה שמרנו את המקומות אליהם נגשנו בהאש, ונבדוק

האם נותרו שם קונקשנים שלא נכתבו ו\או שוחררו. את אלה שגודלם עונה על הנדרש מוידאו נשלח

לכתיבה, ואת השחרור נבצע לכולם.

היתרון ברשימה כזו על פני מעבר על טבלת ההאש כולה, הוא במקרים בהם מרבית הטבלה לא היתה

בשימוש, והמידע התנקז ברובו לאותם מקומות. בכך נחסוך את המעבר על השטח המיותר שעשוי

להיות גדול בהרבה מהמנוצל. אמנם אנו מוסיפים פעולה של הכנסת מצביע לרשימה בכל יצירת רשימה

חדשה של קונקשנים, אבל כיוון שהדבר יקרה רק בגישה הראשונה לכל תא, יוצא שבסיכום סופי לא

נעלה את הסיבוכיות. או במילים אחרות, במקרה הטוב נועיל ובמקרה הגרוע לא נזיק...

כשסיימנו את הטיפול בקונקשנים והטרנזקציות כולם, נכתוב את סטטיסטיקות הvideo שאגרנו

במהלכם לקובץCsv נפרד. חישוב הסטטיסטיקות הללו נעשה בשני שלבים: שלב א' – בעת כתיבת

הטרנזקציות של כל קונקשן, נוסיף למשתנים המתאימים את הגודל, משך הזמן ומספר הטרנזקציות

שלו, וכן נסכום את הפרשי הזמן בין הטרנזקציות הנ"ל ונוסיף את התוצאה למשתנה הגלובלי. בנוסף

נעדכן את המשתנה sum\_conn הסוכם עבורנו את מספר הקונקשנים הראויים לכתיבה כוידאו.

שלב ב': בסיום השלב הקודם נמצאים בידינו סכומי הזמן, הגודל, כמות הטרנזקציות, הפרשי הזמנים

של הטרנזקציות וכמות של סך הקונקשנים הרלוונטיים עבורינו. כעת נוכל לדעת את הסטטיסטיקות

הבאות: משך הזמן הממוצע של טרנזקציה בוידאו (מתקבל מחלוקת סכום זמני הקונקשנים בסכום

כמות הטרנזקציות), את הפרש הזמן הממוצע ביניהן (סכום ההפרשים לחלק למספר הטרנזקציות

פחות מספר הקונקשנים, שהרי מהטרנזקציה האחרונה בכל קונקשן לא נוצר הפרש זמן עד לטרנזקציה

הבאה – שלא באה:) את הגודל הממוצע לקונקשן ואת משך הזמן הממוצע (סכומי הגודל והזמן לחלק

למספר הקונקשנים), את כמות הטרנזקציות הממוצע (סכום הטרנזקציות חלקי הנ"ל), ואת גודל

הטרנזקציה הממוצע בוידאו (גודל הקונקשן הממוצע חלקי כמות הטרנזקציות הממוצעת).

הערות:

בחרתי להשתמש בפונקציות inline במקומות בהם ידעתי שהפונקציה אמורה להקרא פעמים רבות

במהלך התכנית, כדי לחסוך פתיחת קריאות ולמנוע דליפת זכרון.

בנוסף חשבתי על רעיונות נוספים לשפור הביצועים, כמו שימוש בעצי AVL כתחליף לרשימות בטבלת

ההאש – כך שבמקרה הגרוע הגישה תהיה בO(lgn) ולא בO(n). בחרתי להמנע מכך בשל היתרון

שמאפשרת לי הרשימה – ניצול עקרון הלוקליות, וצמצום הגישה בפרקי זמן סמוכים ל-O(1) (ברוב

מוחלט של המקרים. יתכן שבקלט שונה, אולי בלייב, היה הדבר מניב תוצאות אחרות, אבל במצב

הנוכחי אין ספק שכך עדיף.)

רעיון נוסף שנהגה אך לא מומש היה חלוקת העבודה לthreads, במודל Boss – Worker. גם כאן בחרתי

להמנע, מסיבה שהתועלת המרבית שיכולתי להפיק מכך היתה חלוקת זמן העבודה כמספר הליבות

במחשב לכל היותר, מספר שבמחשב רגיל אינו עולה על 6-8. לעומת זאת היו נדרשות לכל פקטה

פעולות נוספות כמו הכנסה והוצאה מתור המשימות, נעילת ושחרור mutexes וכדומה, כך שהחסכון

בסופו של דבר אינו משמעותי ועלול אפילו להגרע מהתוספת (אם נחלק למשל ב-6 ונכפיל ב-7...).

תודה על המסירות והאכפתיות לאורך כל הדרך, בזמנים עמוסים ואפילו מחוץ לשעות העבודה,

תמיד במאור פנים ובסבלנות גם לשאלות הכי קטנות... למדתי גם בחלק המקצועי וגם בחלק האנושי😊

תודה רבה על הכל!!

חוי פישר