SYN FLOOD

אדם סינאלה ואליהו פרידמן

https://docs.google.com/document/d/1Xo7Ce6cKyE7l9NrWykLe-ChTSN2rsFCCd25p1Q4ilvY/edit?usp=sharing

1. מבוא

התקפת SYN Flood היא סוג של התקפת מניעת שירות (DoS), שבה התוקף מציף את השרת בבקשות TCP SYN מבלי להשלים את התהליך. מטרת התקפה זו היא ליצור עומס רב על השרת באמצעות פתיחת חיבורים רבים מבלי לסגור אותם, מה שמוביל לשימוש מוגבר במשאבי המערכת עד כדי השבתת השירותים.

במטלה זו, נדרשנו ליישם סימולציה של התקפת SYN Flood בשתי שפות תכנות: C ו-Python. בנוסף, התבקשנו לעקוב אחר ביצועי המערכת הנתקפת באמצעות בקשות Ping שנשלחות ממחשב נפרד המשמש לניטור (Monitor). המטלה כוללת השוואה בין הביצועים של שתי השפות ויצירת גרפים המתארים את זמני שליחת הבקשות ואת השפעת ההתקפה על זמן תגובה של שרת היעד.

2. מימוש

1 Attack.c – התקפת SYN בשפת C

Attack.c אחראי על יישום התקפת SYN Flood בשפת C. התוכנית עושה שימוש בRaw Socket שמאפשר שליחה ישירה של חבילות TCP תוך שליטה מלאה על כל שדות החבילה. המערכת יוצרת חבילות SYN ושולחת אותן לשרת היעד בצורה מחזורית, כאשר כל חבילה נשלחת מכתובת IP אקראית כדי לדמות התקפה רחבת היקף ממקורות שונים. התוכנית מודדת את זמן השליחה של כל חבילה ושומרת את התוצאות בקובץ.

2 Attack.py – התקפת SYN בשפת Python

הקובץ Attack.py מיישם התקפת SYN Flood בשפה Python, תוך שימוש בספריית Scapy. כמו בגרסה של C, התוכנית שולחת חבילות SYN לשרת היעד, כאשר כתובת ה-IP של המקור משתנה בכל איטרציה כדי לדמות התקפה ממקורות שונים. התוכנית מודדת את זמן השליחה של כל חבילה ושומרת את התוצאות בקובץ.

2.3 monitor.c – ניטור ההתקפה באמצעות Ping

הקובץ monitor.c אחראי על ניטור הביצועים של שרת היעד במהלך התקפת SYN Flood. התוכנית שולחת בקשות Ping לשרת היעד כל 5 שניות, ורושמת את זמני התגובה (RTT) בקובץ תוצאות. מטרת המעקב היא לבדוק את השפעת ההתקפה על זמני התגובה של השרת.

2.4 graphs.py – יצירת גרפים להצגת תוצאות ההתקפה

הקובץ graphs.py משמש לקריאה של קבצי התוצאות שנוצרו במהלך הניסוי, הן עבור התקפת ה-SYN והן עבור הניטור. באמצעות ספריית Matplotlib, התוכנית יוצרת גרפים שממחישים את הביצועים של ההתקפה, כולל זמני שליחה של החבילות וזמני תגובת השרת במהלך ההתקפה.

2.5 docker-compose.yml – ניהול הסביבה עם Docker

הקובץ docker-compose.yml מגדיר את הסביבה המבוזרת שבה מתבצעת ההתקפה באמצעות Docker. הסביבה כוללת מכולות (Containers) של תוקף, שרת יעד(שרת Apache), ומכולת ניטור. כל מכולה פועלת בצורה עצמאית ומבצעת את התפקיד שלה בסביבה מבודדת.

3. תוצאות והבנה

3.1 ביצועי שליחת חבילות SYN

הגרפים המוצגים מראים את זמני השליחה של חבילות ה-SYN שנשלחו על ידי תוכניות ה-C וה-Python. ניתוח התוצאות מראה שביצועי התוכנית בשפת C היו יעילים ומהירים יותר בהשוואה לגרסה בשפת Python, כתוצאה מהיעילות של שפת C בעיבוד ובשליחת חבילות ברמת מערכת ההפעלה.

3.2 זמני תגובת Ping (RTT)

במהלך ההתקפה נשלחו בקשות Ping לשרת היעד כל 5 שניות, והגרפים מציגים את זמני התגובה של השרת במהלך ההתקפה. ניתן לראות שזמני התגובה של השרת התארכו בזמן ההתקפה, מה שמעיד על העומס שנוצר.

3.3 הבנה וניתוח ההבדלים

השוואת ביצועי התקפת SYN בשפות C ו-Python מראה שהגרסה ב-C פועלת בצורה מהירה יותר. ב-Python הביצועים איטיים יותר.

4. סיכום

במטלה זו יישמנו התקפת SYN Flood בשתי שפות תכנות שונות, C ו-Python, וביצענו ניטור בזמן אמת לביצועי השרת המותקף. התוצאות הראו שהגרסה בשפת C הייתה יעילה יותר מבחינת ביצועים. למדנו נושא חדש, Docker ובאמצעותו ייצרנו סביבה מבודדת ומנוהלת שביצעה את ההתקפה בצורה מבוקרת וניתחה את התוצאות בצורה ברורה.