נושאים במערכות הגנה לרשת

מימוש שרת מייל אשר מספק הגנה ללקוח

שם: דורון בוקובזה, ת"ז: 204274377
שם: אסף מורנו, ת"ז: 311318240

תוכן עניינים

מה הכלי עושה:

בהתאם לתוכנית העבודה בנינו שרת מייל אשר מספק העברת נתונים בטוחה בין הלקוחות.
נציין כי פעולות ההגנה מבוצעות בשרת זאת בכדי שניתן יהיה לשלוח הודעה אזהרה ללקוח המקבל ובמידת הצורך לחסום הודעות אשר מכילות קבצים זדוניים.
הגנה שאנו מבצעים מתבצעת על ידי:

- חסימת קבצים זדוניים אשר עלולים לפגוע במחשב (יזוהו לפי מאגר נתונים).
- זיהוי קישורים פיקטיביים אשר נועדו לדלות פרטים אישיים מהלקוח (יזוהו לפי מאגר נתונים).
- חסימת משתמשים אשר מספימים את השרת (שולחים כמות גדולה של הודעות בהפרשי זמן קטנים).
- זיהוי קבצים אשר עלולים להיות מסוכנים והדפסת הודעת אזהרה על כך (קבצים מסוג exec)
- זיהוי משתמשים בעייתיים (שלחו בעבר הודעות וירוס או ספאם) והדפסת הודעה מתאימה.
- זיהוי הודעות אשר עלולות להכיל תוכן שיווקי (לא נכלל בתוכנית העבודה).

פירוט:

חסימת קבצים זדוניים:

בכדי לזהות קבצים חשודים היה עלינו ליצור רשימה של חתימות מסוימות של ווירוסים מוכרים אשר במידה ואחת מהחתימות הללו יופיעו בקובץ נסמנו כ-"קובץ זדוני" ולכן השתמשנו ברשימה אשר ניתנה בקורס "ארכיטקטורה" בשם "signatures " ובו החתימות של הווירוסים הופיעו ברצף של תווים והופיעו לפי הפורמט:

offset	size (in bytes)	description
0	2	The virus's signature length N, up to 2^16 little endian
2	16	The virus name represented as a null terminated string
18	N	The virus signature

לאחר שהיה לנו קובץ המכיל חתימות של ווירוסים יצרנו מערך נתונים אשר מכיל את כל הווירוסים בקובץ אשר בעזרתו נוודא את האמינות של קבצים נכנסים, כלומר בעת הגעת קובץ ווידאנו כי הוא לא מכיל את אחת מחתימות הווירוסים ממערך הנתונים שיצרנו ובמידה וכן המייל נחסם עם הדפסת הודעה מתאימה בשרת.

קשיים איתם התמודדנו:

בעת העבודה על הפיצ'ר הנ"ל נתקלנו בקשיים בעקבות פירסור נכון של הקבצים אותם קיבלנו בשרת ובדיקתם לפי המערך נתונים אך בסופו הצלחנו לבצע זאת על ידי אפשרות של decode בפונקציה get_payload.

זיהוי קישורים פיקטיביים:

פיצ'ר נוסף שרצינו להוסיף הינו זיהוי של אתרים מתחזים אשר נועדו לדלות פרטים מהלקוח ולכן חשבנו איך יהיה ניתן לזהות אתרים אלו. הגענו למסקנה שבדומה לווירוסים ניתן לתחזק גם כן מאגר של אתרים פיקטיביים ובאמצעותו נאתר קישורים אלו בהודעות המועברות בשרת, ואכן ביצענו זאת על ידי בדיקה של הטקסט המועבר במייל ובדיקה האם הוא מכיל את אחד מהאתרים הללו.
במידה וכן, מודפסת הודעת אזהרה אשר מופיעה בשרת (ניתן לשלוח הודעה זו ללקוח במידה ונדרש).

חסימת ספאמרים:

כפי שהסברנו אנו רוצים למנוע הספאמה על השרת ולכן היה עלינו לזהות מצבים בהם לקוח מסוים מנסה להספים, לשם כך החלטנו כי במידה ומשתמש שולח יותר מ-10 הודעות בדקה הוא יוגדר כמספים (ניתן לשנות זאת בקלות בקוד לפי המשנים הגלובליים : $spam\_period, spam\_count$) ולכן מנענו את האפשרות לכך (כלומר משתמש לא יוכל לשלוח יותר מ – 10 הודעות בדקה)

קשיים:

בדיקה של כמות ההודעות ששולח מסוים שולח בזמן מסוים בצורה יעילה:

תחילה מימשנו זאת על ידי רשימה של מיילים שהולכת וגדלה עם כל מייל שמתקבל ובדיקה של כל הרשימה בסדר יורד אך הבנו שהדבר עלול לגרום לגדילה לא חסומה של הרשימה ולכן חיפשנו דרך חלופית,
לבסוף מימשנו מערך חסום של מיילים אשר מתמלא בצורה מעגלית ושומר לכל היותר מספר סופי של המיילים האחרונים שהתקבלו (בחרנו לשמור כ – 100 מיילים אחרונים אך ניתן לשנות זאת בקוד לפי המשתנה הגלובלי re_size) ובכך קיבלנו את הנדרש.

פיצ'ר EXEC:

בנוסף רצינו גם כן להתריע על קבצים אשר עלולים להיות קבצים בעייתיים במידה והם קבצי $exec$ ולכן היה עלינו לזהות האם הקובץ אשר הועבר הוא אכן קובץ $exec$ ובמידה וכן ברוב המקרים (רק קבצים בפורמט ELF) הדפסנו הודעה מתאימה אצל השרת אשר זיהה זאת.

קשיים:

כיצד לבדוק האם קובץ הוא executable או לא:

תחילה מצאנו פונקציה שבודקת זאת עבור קובץ בדיסק ולכן יצרנו קובץ חדש בדיסק אליו העתקנו את תוכן הקובץ אשר התקבל במייל, אך למרות שהעתקנו את הקובץ נוכחנו לגלות כי סוג הקובץ לא מועבר על ידי הכתיבה לקובץ לפי הדרך בה בחרנו לכתוב לקובץ החדש (($file.write()$),
לאחר מכן נזכרנו שרוב קבצי ה- $executable$ מיוצגים בפורמט ELF ולכן חלק מהבייטים הראשונים בקובץ הם קידוד ה - $ascii$ של "ELF" ולכן בדקנו את הבייטים הרלוונטיים.

זיהוי משתמשים בעייתיים:

כחלק מתהליך ההגנה אשר אנו רוצים לספק אנו רוצים לסמן לקוחות אשר בעייתיים אשר שלחו למשל וירוס או הספימו את השרת ולטפל בהם בהתאם במידה וירצו לשלוח מייל נוסף בשנית,
אנו בחרנו לא לחסום לקוחות אלו לצמיתות אלא לשלוח הודעת אזהרה לגביהם אשר מדובר בלקוח בעייתי ויש לקחת זאת בחשבון.

קשיים

כיצד לשמור משתמשים בעייתיים:

תחילה שמרנו רשימה של אובייקטים מסוג $client$ (שולח) שהמזהה שלהם הוא ה-ip, אך הבנו ששולח יכול לשלוח מיילים מכתובות ip שונות עם אותו המייל ובכך להמשיך בפעילותו הזדונית מבלי שנדע שהוא לא אמין,
ולכן יצרנו רשימה של $ip's$ ומיילים עבור שולחים לא אמינים ובכל פעם ששולח שלח וירוס או ספאם הוספנו לרשימה גם את ה-ip וגם את כתובת המייל שלו. בנוסף, לפני שהוספנו כל אחד מהאיברים לרשימה, בדקנו שהוא לא נמצא בה וכך שמרנו על הרשימה קטנה ככל הניתן.

זיהוי הודעות אשר עלולות להכיל תוכן שיווקי:

אפשרות נוספת אשר חשבנו עליה הייתה לזהות מיילים אשר עלולים להכיל תוכן שיווקי. לכן החלטנו לבצע זאת לפי מאגר של שמות אתרי מכירות נפוצים ובדיקה האם מייל השולח מכיל את אחד השמות הללו.
במידה וכן הודעה זו מסומנת כאופציונלית להודעה המכילה תוכן שיווקי ובשל כך מודפסת הודעה בשרת.

בדיקות:

בכדי לבדוק את השרת היה עלינו לדמות סוגים שונים של לקוחות ולראות שההתנהגות המצופה לגבי כל אחד מהם אכן מתקבלת ולכן יצרנו את הלקוחות לפי השמות:

- client – זהו לקוח סטנדרטי אשר שולח הודעה רגילה אשר לא מצריכה את התערבות השרת בהגנת הלקוח המקבל.
- client_with_virus – זהו לקוח אשר שולח קובץ זדוני אשר על השרת לחסום הודעה זו. גם עבור מקרה זה השתמשנו בקובץ אשר ניתן לנו בקורס "ארכיטקטורה" בשם – "infected ".
- client_fakeLink – זהו לקוח אשר המייל אותו הוא שולח מכיל בתוכו קישור פיקטיבי ולכן על השרת להדפיס הודעת אזהרה על כך.
- client_spam – זהו לקוח אשר מספים את השרת על ידי שליחה של יותר מ-10 מיילים בפחות מדקה וזהו כאמור מצב אשר מוגדר כ-"הספמה" של השרת ולכן מצופה מהשרת לחסום משתמש זה מלשלוח יותר מ-10 מיילים לדקה.
- client_exec – זהו לקוח אשר שולח קובץ מסוג "exec" אך הוא לא בהכרח מציין זאת (בעת שליחת קובץ קיימת אפשרות לשולח לציין את שם הקובץ כולל סיומת ובכך "לזייף" את שם הקובץ) ולכן מצופה מהשרת לציין כי זהו קובץ אשר עלול להיות בעייתי.
- client_saleWeb – זהו לקוח אשר שולח הודעה המכילה תוכן שיווקי ונזהה זאת כפי שציינו לפי המייל של השולח.

חוזקות וחולשות של הכלי:

חוזקות:

- הגנה שקטה – אין צורך בביצוע פעולות מצד המשתמש והוא מקבל את ההגנה בצורה סמויה כאשר כל מייל עובר בדיקה לפני הצגתנו למשתמש.
- אפשרות מהירה לעדכון – מכיוון שאנו עובדים על מידע קיים על הווירוסים הקיימים ברשת אנו יכולים בקלות לעדכן רשימה זו ובכך לקבל הגנה גם עבור קבצים חדשים אשר נתפסו ,זאת בדומה לתוכניות הגנה הקיימות כיום בשוק.

חולשות:

- הגנה לא הרמטית – עדיין קיימים ווירוסים אשר לא מופיעים ברשימת הווירוסים המוכרים לשרת ולכן הגנה זו לא תזהה וירוסים אלו.
- עדכון קבצי השרת – במהלך כתיבת הקוד החלטנו לשמור את הפרטים של הווירוסים הידועים על השרת (לפרסר את הקובץ בתחילת ההרצה ולשמור במערך נתונים) ובכך לחסוך בזמן יקר על ידי פעולה אחת יקרה בהפעלת השרת אך בשל כך במידה ונרצה לעדכן את הקובץ נתונים של הווירוסים יהיה עלינו לאתחל מחדש את השרת ובזאת כל הנתונים על המשתמשים הבעייתיים נמחק אך נציין שניתן לפתור זאת על ידי שמירת הנתונים הללו על הדיסק לפני האתחול ושחזורם לאחר מכן.

תיאור עבודת הכלי:

השרת אשר בנינו מיועד לעבודה מול מערכות לינוקס ומופעל בצורה הבאה:

- בכדי להפעיל את השרת יש להיכנס לתיקייה "project\server"
- יש לפתוח טרמינל באותה תיקייה ולהריץ את הקובץ server.py על ידי הרצת השורה: python3 server.py.
- לאחר מכן ניתן להריץ כל אחד מין הקבצים המסמלצים התנהגות מסוימת של לקוח, אשר נמצאים בתיקיה "project\client" ושמותיהם פורטו קודם לכן בתיאור הבדיקות שבוצעו, ולאחר מכן לצפות בהודעה אשר מתקבלת אצל השרת.