שם המגיש: עומר גרייף

ת"ז : 2082000154

תאריך הגשה: 8/5/23

שאלה 1 – התקפות על רשתות ונוזקות

1. ARP Spoofing –

- מנגנון ARP מאפשר לקבל עבור כתובת IP את כתובת הMAC המתאימה ברשת המקומית. בהתקפה זו התוקף מנצל את הצורך בתרגום כתובת הIP של מכשיר לכתובת MAC בדרך אל מכשיר היעד על ידי הפצת תשובות כוזבות של כתובת MAC בשילוב הIP המבוקש וכך בעצם התוקף מאפשר "פתירה" של הIP הנתון אך עם כתובת MAC שקרית שאליה התוקף מעוניין שהפקטות שנשלחות יועברו. בדרך זו התוקף "מתחזה" למחשב שהפקטה אמורה לעבור דרכו או אליו וכך מתאפשרת לו גישה להליך התקשורת אף על פי שהוא אינו המחשב שאמור לקבל את הפקטה.

- אם התוקף לא יהיה הראשון שיענה על שאילתת ARP כבר הפתרון לIP ידוע והמנגנון מתעלם משאר הפתרונות שמגיעים לאחר מכאן. בפעם הבאה שיהיה צורך לתרגם את הIP כבר לא תצטרך להיות מופצת בקשת broadcast לכל המחשבים ברשת. לכן התוקף נדרש להיות הראשון שעונה לשאילתת ARP. מכיוון שהנחנו שהתוקף יודע את הכתובת שצפויה לבצע את השאילתה התוקף יכול לשלוח הודעות חוזרות ונשנות לפתרון לשאילת זו עם כתובת הIP של הכתובת שמבוצעת עליה השאילתה בשילוב כתוב הMAC של התוקף. התוקף נדרש לשלוח זאת בתדירות גבוהה על מנת לאפשר שבזמן רנדומלי שהשאילתה תתבצע הוא יהיה הראשון לטעון לתרגום.

- כדי להתגונן מהתקפות ARP spoofing דרך אפשרית עבור מנהל הרשת היא לא להסתפק רק בהודעת פתירה הראשונה שמתקבלת ממחשב כלשהו עבור IP המבוקש אלה לתפוס את כל הודעות ההזדהות של מחשבים שמחזירים את כתובת הMAC עבור IP מסוים ועל ידי השוואה בין הMACים המתקבלים ניתן לחשוד בסבירות גבוהה שאם מתקיימת סתירה עבור תרגום IP ליותר מכתובת MAC אחת שאכן מצבעת הטעיה עבור כתובת זו. במקרה זה ניתן להחליט על אי פתירת IP זה , בקשות נוספות לתרגום ועוד.

2. חשיבות התקנת עדכוני אבטחה עבור כלל המשתמשים :

כאשר מתפרסמים עדכוני אבטחה הם באים על מנת לשפר נקודות חולשה של האפליקציה הרלוונטית. במקרה זה על ידי פירוט הבעיות שתוקנו ו/או חקירה של העדכון החדש התוקף יכול לפתח נוזקה שתנצל את חולשה זו אצל משתמשים שעדיין לא ביצעו את העדכון ועל ידי כך לחדור אל המשתמש ו/או אל הגורם המפעיל את האפליקציה.

ואכן כפי שראינו בתרגול את פעולת תולעת blaster שניצלה את פרסומה של מייקרוסופט על עדכון אבטחה MS03-26 שבתיאור שלו הכוון לכך שהייתה חולשה מטיפוס חריגה מחוצץ. התוקף ניצל את המידע של מייקרוסופט על עדכון הגרסה ואת כך שהשירות היה פתוח כברירת מחדל על פורט 445 וכך דאגה לשלוח הודעה שתחרוג מחוצץ המחסנית (כפי שפורסם בעדכון התוכנה) וגרמה לכך שכתובת החזרה תשתנה. ההודעה שהועברה שתלה קוד זדוני על המחסנית וכך גרמה לפגיעה במשתמשים.

3. **<u>תולעת האינטרנט</u>**:

- באותו תאריך בMIT הופעלה תולעת האינטרנט שגרמה לנפילת מחשבים רבים עקב "סתימת" טבלאות תהליכים או שטחי הדפדוף שלהם. הסטודנט שמתואר חווה את השפעת התולעת ואכן עומס התהליכים הקריס את המחשב.

- לאחר הדלקה מחדש של המחשב לקח זמן עד שהמחשב קרס בשנית מכיוון שהתולעת לא השאירה חתימה ובשלב הראשוני תפקוד המחשב היה רגיל. בהמשך המחשב נדבק בשנית עד ששוב התמלאו טבלאות התהליכים ושטחי הדפדוף שהובילו לקריסה בשנית של המחשב, וחוזר חלילה.

- על מנת למנוע את קריסת המחשב הסטודנט יכול היה להתנתק מהרשת שדרכה הופצה התולעת ולמנוע את העמסת המחשב (הרי הוא אינו הזדקק לשירותי הרשת והוא יכול היה לעבוד במצב OFLINE). אם המחשב היה עדיין קורס סימן שהתולעת כבר הדביקה את המחשב ובעת הדלקה חוזרת של המחשב הוא היה יכול לא להתחבר לרשת מלכתחילה וכך בכל זאת היה מצליח להמשיך לעבוד על התרגיל מכיוון שהתולעת נעלמת לאחר הדלקה נוספת.

4. **<u>Buffer Overflow:</u>**

- חולשה זו מנצלת את הצורך לשמור מידע בבאפר כלשהו כאשר גודל הבאפר מוגדר. במקרה זה על ידי דחיפה של הרבה מידע על הבאפר ניצן לחרוג מהגודל שהוקצה לבאפר ולפגוע בשאר המידע ששמור על המחסנית וכך להנדס את המצב שיתבצעו דברים שהקורבן לא ציפה שיקרו ו/או לשתול פקודות מסוימות כפי שמימשנו בקורס את"מ.

- דוגמאות שראינו – תולעת האינטרנט – (שהוסברה בסעיף 3) ותולעת blaster

- תולעת bluster – תולעת זו ניצלה את חולשת האבטחה של מיקרוספט ו"בהכוונתם" לכך שישנה בעיית אבטחה בbuffer מסוים. התוקפים ניצלו את המצב שהרבה משתמשים עדיין לא עדכנו את גרסאות המערכת שלהם ועל כן היו חשופים לפרצת אבטחה זו. התוקפים יצרו תולעת שפנתה לפורט שהיה פתוח כברירת מחדל ושלחה הודעה מהונדסת כך שתחרוג מחוצץ שקיים על המחסנית ותשתול קוד זדוני עליו כך שהוא יתבצע במקום הפעולה התקינה של המערכת.

5. **תיקון מבוקש:**

תמיכה במנגנון התאבדות של התולעת במידה וקיימת כבר תולעת במחשב בה היא קיימת. ניתן לבצע זאת על ידי עדכון הקוד שהתולעת מריצה במחשב ובכך לאפשר שבממוצע תהא רק עותק אחד של תולעת שרץ באותו מחשב.

6. **תקיפת syn attack:**

התקפה זו היא התקפת DDOS למניעת שירות המתבסת על פרוטוקול TCP . בתקיפה זו התוקף שולח מספר רב של חבילות syn ראשוניות ליצירת התקשרות אל הקורבן שגורמת לקורבן לפתוח sessions חדשים אצלו והמתנה לack מהתוקף. על ידי שליחה מרובה של בקשות וללא הודעת ack מהתוקף למותקף מצטברים sessions אצל המותקף עד שכבר לא ניתן לקבל בקשות חדשות כיוון שהמקום המוקצה מוגבל. על ידי כף התוקף פוגע ביכולת של המותקף לקבל בקשות חדשות מלקוחות אמיתיים או לחילופין מצריך ממנו לפנות חלק מהמידע על sessions ישנים שיפגעו במהירות התקשורת של הקורבן. כפי שציינתי הפגיעה מתרכזת במבנה הנתונים ששומר את הsessions הפתוחים ומטרתה למנוע שירות ולהאט את פעולת הקורבן.

שאלה 2 – TCP/ IP

1. **הסבר השכבות :**

מודל השכבות נועד לאפשר לחלק את ההתקשרות לרמות שונות של מרכיבי התקשורת החל מהחומרה ועד שכבת האפליקציה. בעזרת מודל זה ניתן לטפל במידע המועבר בהתאם למאפייני השכבה בעזרת המעטפת והפרוטוקולים של השכבה שהתקבלה ולאחר מכאן לקלף בכל שלב את המעטפת ולעבור לטיפול בשלב של השכבה הבאה עד הגעה לרמת האפליקציה. (כמובן שיש שלבים שנוריד מעטפת ונוסיף אותה שוב עם נתונים שונים לפני שנגיע לרמת האפליקציה כמו למשל במנגנון הARP).

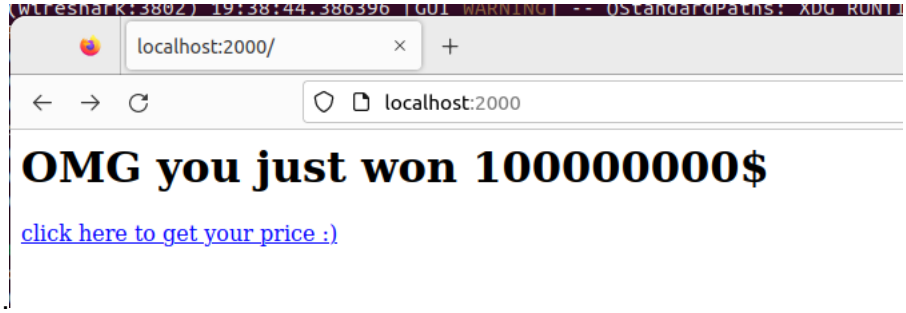
השכבות על פי סדר:

- **השכבה הפיסית –** החומרה – העברת החבילה על בסיס החומרה קרי אותות חשמליים/אופטיים/רדיו. פרוטוקול של העברת אותות חשמליים לדוגמה.
- **שכבת ה-MAC –** אחראית על העברת החבילות בין מחשבים שכנים הנמצאים באותה רשת מקומית. כתובת הMAC היא ייחודית לכל כרטיס רשת בכלל מרחב הרשת. בשלב זה מתבסס על פרוטוקול ethernet.
- **שכבת הרשת/IP –** משמשת להעברת חבילות בין מחשבים רחוקים שנמצאים ברשתות שונות. כל מחשב מקבל כתובת IP ייחודית שבאמצעותה מזוהה המחשב בשלב זה. למשל פרוטוקול IPV4.
- **שכבת התובלה –** לשכבה זו מספר תפקידים מרכזיים: וידוא העברה תקינה של מידע על פי סדר וקצב מתאים (תלוי פרוטוקול) , העברת תקשורת בין אפליקציות של מחשבים שונים ומאפשרת שירותים לכל האפליקציות במכונה. פרוטוקולים בסיסים מוכרים הם UDP (לא אמינה) וTCP (אמינה).

- **שכבת האפליקציה –** האפליקציה- התוכן של ההודעה שלשמה ביצענו את התקשורת בכל השכבות מתחת. מאפשרת את השימוש במידע. למשל פרוטוקול HTTP.

2. **הפלט המבוקש:**

3. **הפלט המבוקש:**



4. **התחברות במקביל client + host :**

הפקודות: 1. ראשית אצל השרת python3 Desktop/server.py 2000
2. שנית אצל המשתמש python3 Desktop/client.py 127.0.0.1 2000
מימין מרחב הרצה של השרת ומשמאל מרחב הרצה של המשתמש:

5. **הסבר פונקציות socket בהן השתמשתי :**

- **Bind –** מחבר את הsocket לפרטי השרת הרלוונטים לתקשורת – port ו ip. (השתמשנו בקוד שרת על מנת להגדיר לו את חיבור הsocket מצד השרת)
- **listen –** מאזין על הפורט עד לקבלת התקשרות עם לקוח. (שימוש בקוד שרת לאחר יצירת הsocket לפני התחברות הלקוח)

- **connect –** מבקש חיבור מצד הלקוח לשרת על ידי בקשת "חיבור" הsocket בצד הלקוח עם פרטי הsocket של השרת (כפי שהוסבר שימוש בצד לקוח)
- **accept –** מאשר את קבלת הבקשה לחיבור מהלקוח אצל השרת לאחר המתנה ומאפשר יצירת תקשורת מעתה ואליך (שימוש בצד שרת לאחר המתנה ולפני המשך תקשורת)
- **send/recv –** פקודות לצרכי העברת המידע בין שני הצדדים – לקוח ושרת המאפשרות העברת פקטות וקבלת פקטות על גבי הsocket שנפתח . send לשליחת פקטה וrecv לקבלת פקטה. (שימוש בשני הצדדים כפי שהוסבר)
-

שאלה 3 – חומות אש

1. **התחברות של עובד בתל אביב לשרתי DB בחיפה:**

ראשית נבחין כי הכלל הדיפולטי מאפשר לכללים שאינם מופיעים להתבצע בשונה מעקרון המינימליות של הכללים שבדרך כלל נהוג שמגדיר את הדיפולטי כדחייה. לכן אם לא ניפול תחת הקטגוריה של אף כלל נוכל להעביר את החבילה. על כן אם נבחר פורט אצל עובד בתל אביב קטן מ1024 לא ניפול תחת אף קריטריון למעט הדיפולטי ונצליח לתקשר עם פורט 80 של DB בחיפה. כדי להשתמש בפורטים אלו עליו להיות בעל הרשאות root.

2. **טבלת חוקים עדכנית:**

rule	direction	Src adrds	Dst adrrs	Next protocol	Src port	Dst port	ACK	action
Spoof_out	out	Any/TLV	Any	Any	Any	Any	Any	deny
Spoof_in	in	TLV	Any	Any	Any	Any	Any	deny
DB_out	out	M	DB	TCP	>1023	80	Any	allow
DB_in	in	DB	M	TCP	80	>1023	Any	allow
Http_out	out	TLV	Any/DB	TCP	>1023	80	Any	allow
Http_in	in	Any/DB	TLV	TCP	80	>1023	Yes	allow
default	Any	Any	Any	Any	Any	Any	Any	deny

3. **התקפות על בסיס ההגדרות הנתונות:**

- העלאת שכר – על ידי ip_Spoofing העובד יכול לשלוח פקטה שמתאימה להעלאת השכר שלו (על פי ההגדרות הנתונות (1 ID AMOUNT) על ידי הגדרת הפקטה כך שהיא נראת כאילו היא נשלחה מ IP של מנהל ולא הIP האמיתי שלו. מכיוון שהעובד נמצא לפני חומת האש בצד של TLV הוא יצליח ליפול בקרטריון שמאפשר תקשורת עם DB בהתחשב בכך שהוא אכן ביצע spoofing IP לIP של המנהל. וכמובן שהוא

לא יפול בקרטריון IP spoofing מכיוון שהוא שולח מTLV ועל כן מבחינת חומת האש הכל נראה תקין.

- העלאת דרגה - כעת בגלל שנדרשת לחיצת יד משולשת נגרום לARP spoofing כך שניתוב הפקטות לאחד המנהלים יגיע אל כתובת הmac של העובד המעוניין בהעלאת דרגה. ניתן לבצע זאת למשל על ידי הצפת טבלאות הניתוב של ARP כך שיזרקו פתרונות ישנים. לאחר מכאן נבצע IP spoofing לכתובת של מנהל כלשהו ובמקביל נשדר לARP שאנחנו (כתובת הMAC של התוקף) זהו הפתרון לIP זה. מיד לאחר מכאן נוכל לבצע התקשרות עם DB תוך התחזות לIP של המנהל ונקבל את הפקטות המיועדות אליו לצורך "לחיצת יד משולשת" וביצוע בקשת העלאת שכר.