

אבטחת מחשבים ורשתות תקשורת

(372-1-4601)

מטלה מס' 3

מתרגל אחראי: חן דויטשמן

הנחיות כלליות:

1. העבודה מורכבת משני חלקים – חלק תיאוראי וחלק מעשי.
2. שאלות על העבודה יש לשאול אך ורק בפורום הייעודי ב-Moodle. שאלות אשר ישאלו בדוא"ל לא יענו.
3. על התשובות להיות מלאות ומפורטות.
4. כחלק מתהליך בדיקת העבודה תתבצע בדיקה לזיהוי עבודות מועתקות. כל מקרה של העתקה יטופל ע"י ועדת משמעת אוניברסיטאית.

הוראות הגשה:

1. ההגשה היא בזוגות בלבד.
2. על התשובות שלכם להיות מוקלדות במחשב או סרוקות בכתב יד נקי וברור.
3. יש לענות על השאלות ולהפיק קובץ PDF עם התשובות שלכם.
4. רק אחד מחברי הצוות יגיש את המטלה.
5. שם הקובץ יהיה ID1_ID2.pdf כאשר ID1 ו-ID2 מוחלפים בת.ז של מגישי המטלה.

בהצלחה

חלק תיאורטי (70 נק')

שאלה מס' 1 – Buffer overflow א' (10 נק')

נתונה הפונקציה הבאה הכתובה בשפת c:

```
1 void func () {
2     char* date = getDateString();
3     int dateLength = strlen(date);
4     char b[64];
5     gets(b);
6     char logMessage[128];
7     strcpy(logMessage, date);
8     strcpy(logMessage + dateLength, b);
9     writeLog(logMessage);
10 }
```

כמו כן נתון כי:

- הפונקציה getDateString מחזירה char* המכיל את התאריך ברגע הקריאה לפונקציה.
 - הפונקציה writeLog מקבלת char[] וכותבת את תוכנו לקובץ לוג.
 - הפונקציות strlen, strcpy ו-gets הינן כפי שהן מתוארות בתיעוד של שפת c.
1. (2 נק') מהי מטרת הפונקציה?
 2. (3 נק') הקומפיילר אשר באמצעותו מקומפל קוד זה משתמש ב-Stack Canary בגודל 4 bytes. שרטט/י את המחסנית לאחר הקריאה לפונקציה וביצוע שורות הקוד 1 עד 6 (כולל) – כלול/י השרטוט את כתובת החזרה מהפונקציה.
 3. (7 נק') ידוע שהקוד קומפל תוך שימוש ב-Terminator Canary, לכן מימוש התקפת buffer overflow פשוטה ע"י הכנסת קלט ארוך וניחוש ה-Canary לא יעבוד. בכל זאת ניתן לבצע התקפת buffer overflow ובכך לדרוס את כתובת החזרה מהפונקציה, וזאת מבלי לשנות את ה-Canary. הסבר/י כיצד ניתן לבצע זאת.
 4. (3 נק') תנאי דוגמה לקלט אשר ידרוס את כתובת החזרה (return address) וישנה אותו לערך 0xDEADBEEF אך לא ישנה את ערך ה-Canary.

שאלה מס' 2 – Buffer overflow ב' (10 נק')

נתון הקוד הבא אשר server כלשהו משתמש בו:

```
1 void func () {
2     char[] b[64];
3     printf("Type a Log Message");
4     gets(b);
5     writeLogMessage(b);
6 }
```

כמו כן, נתון כי:

- הקומפיילר אשר באמצעותו מקומפל קוד זה משתמש ב-Random Stack Canary בגודל 4 bytes אשר לא מכיל תווי Termination.
- ה-Canary איננו משתנה בכל ריצה חדשה של הקוד.
- כאשר המערכת מזהה שה-Canary נדרס, ה-server מבצע אתחול מחדש – תהליך אשר לוקח שנייה אחת.

תוקף מעוניין למצוא את ה-Canary של התוכנית.

4. (4 נק') את/ה מנהל/ת אבטחת המידע (CISO) בארגון שלך. התאמ/י לכל תרחיש את סוג ה-firewall הבסיסי ביותר שמקיים את הדרישה. הסבר/י בקצרה את תשובתך.
- מעוניינים לסנן הודעות מייל שעלולות להכיל קישורים ממקור שאינו מוכר לארגון.
 - מעוניינים לסנן הודעות מייל שמגיעות מתוך הארגון אך עם IP מחוץ לארגון.
 - מעוניינים בפתרון אבטחה שלם הכולל anti-malware, סינון URL, sandboxing וכו'.
 - מעוניינים לסנן תעבורת TCP עוינת למטרות DDoS שמגיעה ממדינה עוינת.

שאלה מס' 4 – (10 נק')

ענו על השאלות הבאות בקצרה:

1. (4 נק') קראו על ה-google FLoC והסבירו מה מטרתו. ציינו יתרון אחד וחסרון אחד בהיבט ה-privacy.
2. (5 נק') קראו על ה-SolarWinds hack.
 - א. (1 נק') פרטו בקצרה על התקיפה.
 - ב. (1 נק') באיזו סוג התקפה מדובר?
 - ג. (1 נק') מהו ה-attack vector שבו השתמשו התוקפים?
 - ד. (1 נק') מהו הנזק לו גרמו התוקפים?
 - ה. (1 נק') באילו דרכים הייתם ממליצים להתגונן מפני תקיפות דומות בעתיד?
3. (1 נק') **בונוס:** היכנסו ל-chrome://settings/passwords/check?start=true (הדביקו את הכתובת בשורת הכתובת בדפדפן chrome). החליפו 3 סיסמאות שלכם/ן שזוהו כ-compromised וצרפו צילומי מסך של מייל האישור מהאתר על החלפת הסיסמה.

שאלה מס' 5 – (15 נק')

- בתרגול למדנו כי המחסנית (Stack) גדלה לכיוון הכתובות הנמוכות בזיכרון.
- בשאלה זו נניח כי המחסנית גדלה בכיוון ההפוך (לכיוון הכתובות הגבוהות בזיכרון).
1. (3 נק') איזה מרחב (Space) נמצא מעל למחסנית? מה תפקידו? איך ניתן לפנות אליו מה-User space?
 2. (12 נק') האם ניתן לבצע מתקפת Buffer Overflow במצב זה?
 - a. במידה ולא, פרט/י כיצד ההגנה מתבצעת.
 - b. במידה וכן, פרט/י כיצד התקיפה מתבצעת.
- התייחס/י לכתובות החזרה (RA) ומצביעי המחסנית (ESP,EBP) בתשובתך וספק/י דוגמת קוד רלוונטית אשר תאשש את טענתך.
- התייחס/י לכתובות החזרה (RA) ומצביעי המחסנית (ESP,EBP) בתשובתך וספק/י דוגמת קוד רלוונטית אשר תאשש את טענתך.

שאלה מס' 6 – (10 נק')

ענו על השאלות הבאות בקצרה:

3. (2 נק') הסבר/י בקצרה על ההבדל בין פרוטוקול TCP ו-UDP.
4. (2 נק') הסבר/י בקצרה על מנגנון ה-Three Way Handshake של TCP.
5. (3 נק') הסבר/י בקצרה על התקפת SYN flood. בתשובתך פרט למה משמשת התקיפה וכיצד ניתן למנוע אותה (2 דרכים לפחות).
6. (3 נק') הסבר/י בקצרה על התקפת SYN-ACK.

חלק מעשי (30 נק')

לצורך פתרון תרגיל זה, נשתמש ב-bWAPP (קיצור של Buggy Web Application) אשר הינה אפליקציית web עם חורי אבטחה רבים. **אנא העזרו במדריך setting up bwapp שנמצא במודל** כדי להתקין את הכלים הנדרשים לחלק זה של המטלה.

כעת פתחו מספר משתמשים במערכת כמספר מגישי העבודה (פתיחת משתמש חדש במערכת נעשית תחת הלשונית New User).

לאחר מכן, התחברו למערכת עם אחד מהמשתמשים שפתחתם.

הערה: פתרון מלא לשאלות הבאות צריך לכלול הסברים מלווים בצילומי מסך וקטעי קוד בהם השתמשתם.

שאלה מס' 7 – CSRF (10 נק')

הכנסו לחור האבטחה Cross-Site Request Forgery (Transfer Amount) תחת Choose Your Bug.

לפניכם מערכת להעברת כספים לחשבון אקראי. נסו את המערכת, הזינו מספרי חשבון וסכומים ובדקו שהמערכת מתנהגת בהתאם למצופה.

1. (2 נק') הסבירו בקצרה על מתקפת CSRF.
2. (5 נק') מצאו דרך בה תוקף יכול לגנוב את כל הכסף בחשבוננו של הנתקף ע"י שימוש ב-CSRF. הדגימו את הדרך הזו ע"י שילוב הבקשה בעמוד HTML לגיטימי כלשהו אותו תבנו וודאו שהמתקפה אכן עובדת. תשובה לשאלה זו תהיה סדרת הצעדים (הכנת עמוד ה-Html, הכנת ההודעה בשילוב Social Engineering, והוכחה שהסכום אכן נגנב).
3. (3 נק') כיצד ניתן להתגונן מהמתקפה אותה הצעתם? הציגו 3 דרכים.

שאלה מס' 8 – Reflected XSS (7 נק')

הכנסו לחור האבטחה Cross-Site Scripting – Reflected (GET) תחת Choose Your Bug.

נסו את המערכת, הכניסו נתונים כרצונכם ושימו לב כיצד המערכת מתנהגת.

1. (2 נק') הסבירו בקצרה על מתקפת XSS – Reflected.
2. (3 נק') חשבו על רעיון יצירתי בו אתם רוצים להשיג את כתובת ה-IP+Cookies של המחשב הנתקף. בתשובתכם התייחסו לכל התהליך.
3. (2 נק') ציינו 3 שלבי בדיקה שתבצעו בתור מומחי אבטחת מידע כדי לבדוק שלינק מסוים לא מכיל מתקפת XSS – Reflected.

הערה: הניחו כי האתר אינו מוגן מהמתקפה.

שאלה מס' 9 – Stored XSS (13 נק')

הכנסו לחור האבטחה XSS – Stored (Blog) תחת Choose Your Bug.

לפניכם בלוג בו אתם יכולים להגיב כרצונכם, תגובותיכם נשמרות במערכת כך ששאר המשתמשים יכולים לצפות בהן (בדקו זאת!).

1. (2 נק') הסבירו בקצרה על המקתפה XSS – Stored.
2. (4 נק') כתבו תגובה בבלוג אשר תגרום לכך שכל משתמש אשר יכנס לעמוד זה, יפתח לו חלון אשר מכיל את הטקסט:

“hacked by ITNS21 student.”

ולאחר מכן יחוזה תקיעה משמעותית בדפדפן/קריסה.

הערה: צוות הקורס לא מעודד תקיפה של אתרי אינטרנט ו/או אפליקציות רשת.

3. (7 נק') מצאו דרך בה תוקף עם גישה לעמוד יצליח לקבל את המיקום הגיאוגרפי של המחשב הנתקף ע"י כתיבת תגובה בבלוג. בתשובתך התייחס לפעולות שהתוקף צריך לבצע.

הערה: במידה ובחרתם לעבוד עם bee-box (אופציה מס' 2 במדריך), בצעו את המשימה הבאה במקום:

מצאו דרך בה תוקף עם גישה לעמוד יצליח לגרום לדפדפן של המחשב הנתקף לגלוש לאתר לבחירתכם ע"י כתיבת תגובה בבלוג. על התוקף לקבל חיווי שמעבר הדף הצליח. בתשובתך התייחס לפעולות שהתוקף צריך לבצע.

הערות:

- ניתן לעבוד עם 2 מחשבים הנמצאים מאחורי NAT.
- תוכלו לבדוק את עצמכם ע"י שימוש ב-Wireshark או Netcat.
- פתרון לשאלה זו יהיה קטעי קוד וצילומי מסך של רצף הפעולות שהתוקף מבצע שמוכיח היתכנות של המתקפה.
- נדרש להראות פלט במסך של התוקף עם המיקום הגיאוגרפי של המחשב הנתקף.
- רמז: פה