סמסטר ב', תשע"ג

ת"ז:

אוניברסיטת תל-אביב
מדעי המחשב

מבוא לאבטחת מידע
מועד א' + פתרון

זמן המבחן: 3 שעות

תאריך: 21.7.2013

מרצים: ערן טרומר ואבישי וול

<u>הנחיות:</u>

- מותר להשתמש בכל חומר עזר כתוב על גבי נייר בלבד.
- המבחן כולל 6 שאלות. מספר הנקודות מופיע לידי כל שאלה.
- כתבו את תשובותיכם <u>על גבי טופס המבחן</u> במקום המוקצה לכך.
  <u>מומלץ מאד</u> לכתוב תחילה את התשובה במחברת הטיוטה שקיבלתם ורק אחר כך להעתיק אותה, בצורה ברורה וקריאה, לטופס המבחן. תשובות במחברת לא יקראו.
- נמקו בקצרה אך בבהירות את כל טענותיכם. כל שאלה לא נימוק לא תזכה בניקוד.
- ניתן לענות בעברית או באנגלית
- במבחן זה <u>10</u> עמודים (כולל עמוד זה). אנא ודאו שכולם ברשותכם.

ב ה צ ל ח ה!

לשימוש הבודקים:

1. (א)	2. (א)	3. (א)	4. (א)	5. (א)	6. (א)	
(ב)	(ב)	(ב)	(ב)	(ב)	(ב)	
	(ג)	(ג)		(ג)		
	(ד)	(ד)		(ד)		
		(ה)				
		(ו)				

שאלה 1 (10 נק')

שרת אינטרנט מסויים מציג מסך חיפוש ע"פ שם משפחה, בו יש להזין שם מבוקש. הקוד שמבצע השרת לאחר לחיצה על כפתור "go" פונה ל-database ובאמצעות שאילתא שנבנית כך:

```
$sql = "SELECT lname, fname, phone FROM usertable
        WHERE lname='" + $_GET["lname"] + "';"
```

א. [4 נק'] כאשר מקלידים את השם O'Brian בשדה החיפוש מופיעה בדפדפן הודעת שגיאה. הסבירו את הסיבה לתקלה.

נראה שאין סניטציה למשתנה הקלט lname, ולכן מתבצעת שאילתת ה-SQL

... `where lname='O'Brian';`

הגרש מהקלט סוגר את המחרוזת, ואחריה כתוב `Brian';` וזה אינו תחביר SQL חוקי.

ב. [6 נק'] תנו דוגמא ל-"שם" שיכול להקליד משתמש בעל כוונת זדון כדי להתקיף את המערכת – הניחו כי אין בדיקות תקינות על תוכן השדה.

כמה דוגמאות:

מציאת שם פרטי+משפחה של משתמש לפי ת"ז:

```
xxx' or id='123456789
```

שינוי סיסמת מנהל האתר:

```
'; update usertable set password='123' where name='admin'; --
```

מחיקת הטבלה Student:

```
Robert'); DROP TABLE Student;--
```

שאלה 2 (20 נק')

ארגון ממשלתי חרד מפני דליפת מספר גדול של מסמכים סודיים בו-זמנית. לפיכך, הוחלט להתקין על כל המחשבים האישיים מנגנון חדש בשם "מאורת-שלג" אשר מונע פתיחה של קבצים אשר שמם מכיל את המחרוזת "secret" בקצב של יותר מאחד בשנייה. (כלומר, ברגע שמשתמש פותח קובץ כזה, לא ניתן לפתוח קובץ נוסף כזה למשך השנייה שאחר כך.) כמובן, המנגנון צריך גם למנוע הסרת המחרוזת "secret" על ידי שינוי שם הקובץ.

הנחות: מערכת הקבצים אינה תומכת ב-links. אין התקפות חומרה. למשתמש אין הרשאות root/Administrator. המחשב אותחל, ומערכת ההפעלה נטענה, כפי שהותקנו ע"י הארגון.

א. [5 נק'] האם וכיצד ניתן לממש מאורת-שלג בעזרת access control lists של מערכת הקבצים? נמקו.

לא. למנגנון ה-ACL אין מצב (state) המושפע מפתיחת קבצים, ולכן אינו מספק דרך לזכור את זמן הפתיחה האחרון. כמו כן, הוא אינו מספק דרך להביע התנהגות תלוית-זמן (למעט ע"י תוכנית אשר תשנה את ה-ACL של כל הקבצים הסודיים, אך אין מנגנון אשר יפעיל אותה בזמן הנכון).

ב. [5 נק'] האם וכיצד ניתן לממש מאורת-שלג בעזרת system call interposition? נמקו.

כן. נכתוב reference monitor אשר מנטר גישות לקבצים של כל התהליכים ואוכף את כללי הפתיחה ושינוי השם האמורים, בעזרת רישום מרכזי של הזמן האחרון בו פתחו קובץ סודי. יש להשתמש בפתרון ניטור בטוח (לדוגמה, חסין בפני התקפות race condition בניגוד ל-PTrace).

התעורר חשש שמשתמש זדוני יעקוף את מנגנון מאורת-שלג על ידי איתחול המחשב ממערכת הפעלה אחרת.

ג. [5 נק'] כיצד ניתן לממש מאורת-שלג חסינה מפני איום זה, ומה מגבלות השימושיות של פתרון זה?

נשתמש ב-TPM, ונבצע seal ברמת מערכת הקבצים או ברמת הקבצים הסודיים.

ה-seal יהיה לערכי PCR המייצגים את האתחול, מערכת ההפעלה **ואת קוד האכיפה הממומש כבסעיף ב'**. חסרונות שימושיות: שדרוגי מערכת הפעלה ושינויי חומרים דורשים seal מחדש, ועובד רק על מחשבים עם חומרת TPM .

ד. [5 נק'] הוחלט שהשרת הארגוני יסכים לשלוח מסמכים סודיים למחשבים אישיים רק אם הוא בטוח שהמסמכים יוגנו על ידי מנגנון מאורת-שלג. כיצד ניתן לאכוף זאת?

נשתמש במנגנון attestation בעזרת TPM. השרת ותוכנה על גבי הלקוח יצרו ערוץ מאובטח (כפי שראינו בכיתה) לאחר קבלת quote המעיד שהתוכנה רצה על גבי מערכת ההפעלה + **מערכת האכיפה המקומית מסעיף ג'** (ובפרט מבצעת seal כדי למנוע את התקפה מסעיף ג'), ובנוסף **כותבת את המידע המתקבל לקובץ הכולל "secret" בשמו**.

שאלה 3 (30 נק')

(שימו לב, חלק מהסעיפים ניתנים לפתרון גם אם לא פתרתם את הקודמים להם.)

במהלך בדיקה שגרתית במחשב נתגלה סוס טרויאני. מנהל המערכת הצליח להשיג את קובץ הריצה של הסוס הזדוני וגילה שהוא מתחבר החוצה בפורט TCP מספר 21 (המוקצה ל-FTP).

מנהל הרשת הציע לך להאזין לרשת ולחפש את כל התקשורות בפורט 21 (0x15), אך גילית כמויות אדירות של תעבורה לגיטימית שקשה להבדיל בינה לבין הסוס. לכן נצטרך לבצע זאת בצורה טובה יותר.

א. [4 נק'] הקוד הזדוני גדול מאד. כיצד נאתר את קטע הקוד המעניין לצורך הבנת שיטת התקשורת של הסוס? (הניחו שהקוד קומפל ללינוקס ע"י gcc ללא עיבוד נוסף).

ניתן למצוא ע"י חיפוש ב-IDA של XREF-ים ל-connect שנמצא בimport table.

לאחר החיפוש מצאת הפונקציה הבאה:

```
.text:080484B4                  ; ¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦ S U B R O U T I N E ¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦
.text:080484B4
.text:080484B4                  ; Attributes: bp-based frame
.text:080484B4
.text:080484B4                  sub_80484B4  proc near              ; CODE XREF: .text:080483FFp
.text:080484B4
.text:080484B4                  var_18       = sockaddr ptr -18h
.text:080484B4                  var_4        = dword ptr -4
.text:080484B4
.text:080484B4  55                           push    ebp
.text:080484B5  89 E5                        mov     ebp, esp
.text:080484B7  53                           push    ebx
.text:080484B8  83 EC 18                     sub     esp, 18h        ; Integer Subtraction
.text:080484BB  6A 00                        push    0               ; protocol
.text:080484BD  6A 01                        push    1               ; type
.text:080484BF  6A 02                        push    2               ; family
.text:080484C1  E8 EA FE FF FF               call    _socket         ; Call Procedure
.text:080484C6  83 C4 0C                     add     esp, 0Ch        ; Add
.text:080484C9  66 C7 45 E8 02 00            mov     [ebp+var_18.sa_family], 2
.text:080484CF  66 C7 45 EA 00 15            mov     word ptr [ebp+var_18.sa_data], 1500h
.text:080484D5  C7 45 EC 88 88 34 12         mov     dword ptr [ebp+var_18.sa_data+2], 12348888h
.text:080484DC  6A 10                        push    10h             ; int
.text:080484DE  89 C3                        mov     ebx, eax
.text:080484E0  8B 45 EC                     mov     eax, dword ptr [ebp+var_18.sa_data+2]
.text:080484E3  2D 33 11 33 11               sub     eax, 11331133h  ; Integer Subtraction
.text:080484E8  89 45 EC                     mov     dword ptr [ebp+var_18.sa_data+2], eax
.text:080484EB  8D 45 E8                     lea     eax, [ebp+var_18] ; Load Effective Address
.text:080484EE  50                           push    eax             ; struct sockaddr *
.text:080484EF  53                           push    ebx             ; int
.text:080484F0  E8 CB FE FF FF               call    _connect        ; Call Procedure
.text:080484F5  83 C4 10                     add     esp, 10h        ; Add
.text:080484F8  85 C0                        test    eax, eax        ; Logical Compare
.text:080484FA  74 0F                        jz      short loc_804850B ; Jump if Zero (ZF=1)
.text:080484FC  83 EC 0C                     sub     esp, 0Ch        ; Integer Subtraction
.text:080484FF  53                           push    ebx             ; fildes
.text:08048500  83 CB FF                     or      ebx, 0FFFFFFFFh ; Logical Inclusive OR
.text:08048503  E8 D8 FE FF FF               call    _close          ; Call Procedure
.text:08048508  83 C4 10                     add     esp, 10h        ; Add
.text:0804850B
.text:0804850B                  loc_804850B:                         ; CODE XREF: sub_80484B4+46j
.text:0804850B  89 D8                        mov     eax, ebx
.text:0804850D  8B 5D FC                     mov     ebx, [ebp+var_4]
.text:08048510  C9                           leave                   ; High Level Procedure Exit
.text:08048511  C3                           retn                    ; Return Near from Procedure
.text:08048511                  sub_80484B4  endp
```

ב. [10 נק'] הנדסו לאחור את הפונקציה sub_80484B4 ותארו את עיקר פעולתה.

הפונקציה מייצרת socket, עושה connect ואז (אם אין שגיאה) מחזירה את ה-sockfd.

כתובת ה-IP היא 0x113311330 – x12348888 כלומר 0x010177550.

ג. [4 נק'] כיצד ניתן למצוא את המחשבים הנגועים ע"י ניתור התעבורה ברשת, ללא false positives? ציינו כלי וכיצד להשתמש בו.

ניתן לזהות ע"י חיפוש תקשורת לפורט 21 ולכתובת ה-IP מהסעיף הקודם,
בעזרת WireShark עם פילטר מתאים מופעל על השרת המחבר את הארגון לאינטרנט.

בסוס יש קוד אשר מנצל חולשה מקומית לא מוכרת מסוג stack overflow, המאפשרת לו לקבל הרשאות root.
הסוס משתמש ב-shellcode הבא לניצול החולשה:

```
0000    30 31 32 33 34 35 36 37  38 39 30 31 32 33 34 35    0123456789012345
0010    36 37 38 39 30 31 32 33  34 35 36 37 38 39 61 62    67890123456789ab
0020    63 64 65 66 67 68 69 6A  08 04 85 11 08 04 85 11    cdefghij........
0030    08 04 85 11 08 04 85 11  08 04 85 11 08 04 85 11    ................
0040    08 04 85 11 08 04 85 11  08 04 85 11 08 04 85 11    ................
0050    B7 F0 F4 80 B7 F1 F4 81  B7 F0 B7 F0 F9 E0          ..............
```

הניחו שכתובת המחסנית בתחילת הפעלת התוכנה הוא 0xDEADBEEF. הניחו שה-shellcode שלעיל מספיק לניצול החולשה לפתיחת shell, ללא קוד נוסף. המכונה היא big endian.

ד. [4 נק'] איזה סוג של shellcode זה? נמקו.

ה-shellcode הוא מסוג ROP. ניתן לראות זאת ע"י השימוש ב-RET slide. זאת מתוך
ההנתון שזהו הקוד ההתקפי היחיד, והעובדה שאיננו מצביעים לתוך המחסנית
שכתובתה נמצאת באיזור של 0xDEADBEEF, ולכן אנו יכולים לראות שלא הכתובת של
ה-RET ולא הכתובות ב-shellcode קשורות למחסנית.
בנוסף יש רמז בתרגיל שהכתובת של ה-RET נמצאת בקוד IDA כ-RET בסעיף ב'.

ה. [4 נק'] האם shellcode זה יעבוד כאשר ASLR מופעל, ומדוע?

זה לא יעבוד, כיוון שהכתובות של הספריה דינמיות תחת Address Space Layout
Randomization, ולכן הכתובות שרשמנו אינן רלוונטיות.

ו. [4 נק'] מה אפשר להסיק מן ה-shellcode על מבנה מחסנית התוכנה?

מבנה המחסנית הוא: חוצץ+משתנים+אוגרים שמורים, מעליהם EBP; ומעליו
כתובת החזרה. מאחר וה-shellcode מכיל 40 בתים של ריפוד ואחריהם 40 בתים
של RET slide, נסיק שהמרחק בין תחילת החוצץ לכתובת החזרה הוא בין 40 ל-80
בתים.

שאלה 4 (10 נק')

נתונה הפונקציה הבאה בשפת C:

```
static float latest;
void func(int i, float f1, float f2) {
  float *out = &latest;
  float vec[10];
  if ((i<0) || (i>10)) return;
  vec[i] = f1;
  *out = f2;
}
```

הניחו כי הקוד רץ על מחשב עם ארכיטקטורה של 32 ביט, וכי התוקף מספק את שלושת המספרים המועברים כפרמטרים לפונקציה.

א. [7נק'] הסבירו מדוע הקוד הנ"ל פגיע להתקפת control hijacking – ואיך ההתקפה עובדת.
(ניתן להניח הנחות סבירות, אם מציינים אותן במפורש.)

בקוד יש חולשה מסוג off-by-one של בדיקת המשתנה i. אם נעביר i=10, אזי
X, אזי השורה האחרונה תכתוב את f2 לכתובת f1 (כאשר מפרשים את שני ה-floatים
כמילים בנות 32 בתים, ולא כמספר נקודה צפה). ניתן כך לכתוב מילה לכל
כתובת. לדוגמה, בעזרת מספר קריאות לפונקציה ניתן לטעון ROP shellcode לתוך
המחסנית, ואז (בקריאה נוספת) לשנות את כתובת החזרה כך שתצביע אל ה-shellcode.

ב. [3 נק'] נניח שמריצים את הקוד הנ"ל על מחשב שבו DEP פועל. האם ההתקפה תחסם בעקבות כך?

התקפת ה-ROP הנ"ל לא תחסם, מאחר ורק כתבנו למחסנית כתובות חזרה, ולא ניסינו
להריץ ממנה קוד.
(יש התקפות אחרות שניתן לתאר בסעיף א' שכן יחסמו.)

שאלה 5 (20 נק')

לבנק יש שני אתרים ברשת עבור לקוחותיו: superbank.com לניהול חשבונות
ו-superbank-1plus1.com למבצעי הטבות. בשני האתרים, ההזדהות היא לפי שם משתמש וסיסמה הנבחרים על ידי הלקוח. אתר ההטבות מתוחזק על ידי חברה חיצונית, במערכת נפרדת עם מנגנון ההזדהות נפרד לחלוטין (כולל שמות המשתמש והסיסמאות). יצירת החשבונות ופרטי ההזדהות נעשית בסניף הבנק.

החברה החיצונית מדווחת כדלקמן:

"בשבוע שעבר אתר ההטבות superbank-1plus1.com נפרץ, והמתקיפים השיגו הרשאת Administrator בשרת. אך אל דאגה:

I. איתחלנו את השרת והשווינו את הקבצים ורשימת התהליכים וגיבוי מלפני ההתקפה. הכל מתאים, לכן ברור שההתקפה נעצרה ונחסמה.

II. סיסמאות המשתמשים מאוחסנות באתר בשיטת hash עם salt כנהוג, ולכן אין חשש לגניבת סיסמאות.

III. אפילו אם היו נגנבות סיסמאות, מאחר ומדובר רק באתר ההטבות, המתקיף אולי יוכל להיכנס בחינם לפארקי מים, אך אין כל סכנה לחשבונות לקוחות הבנק."

א. [5 נק'] האם טענה I סבירה? נמקו.

לא. אם הבדיקה נעשית מתוך מערכת ההפעלה, יתכן שמותקן rootkit הגורם
לה לשקר לגבי רשימת התהליכים ותוכן מערכת הקבצים. כמו כן, יתכן שרצים תחת
מכונה וירטואלית זדונית. כמו כן, כנראה שהחולשה עדיין קיימת וניתנת לניצול מחדש.

ב. [5 נק'] האם טענה II סבירה? נמקו.

לא. ה-salt נדרש על השרת לצורך בדיקת ההזדהות. לכן ה-salt כעת ידוע לתוקף.
עבור כל חשבון בנפרד, התוקף יכול להפעיל התקפת מילון. לאור התפלגות
הססמאות השכיחות, אפשר כך לגלות סיסמאות של רבים מהמשתמשים.

ג. [5 נק'] האם טענה III סבירה? נמקו.

לא. סביר שרוב הלקוחות מיחזרו את שם המשתמש והסיסמה שלהם בין שני האתרים.

ד. [5 נק'] מסתבר שיומיים לפני ההתקפה, אחד העובדים בחברת אתר ההטבות קיבל דוא"ל אודות "מסמך חשוב" עם קובץ PDF מצורף. כאשר העובד ניסה לפתוח אותו במחשב הארגוני, Acrobat נסגר מיד. כל מחשבי החברה נמצאים באותו Windows Domain.
תארו תרחיש סביר לאופן בו נפרץ השרת החל מדוא"ל זה, שלב אחרי שלב, תוך שימוש במינוח המקובל לשלבים השונים.

הדואל היה התקפת phishing. קובץ ה-PDF כלל exploit לחולשה ב-Acrobat, אשר ביצע control hijacking כדי להריץ shellcode. הקוד כלל exploit מסוג privilege escalation כדי להשיג הרשאת Administrator . הקוד אז חיפש (או איפשר למתקיף לחפש) את השרת ברשת המקומית (lateral movement), וביצע התקפת pass the hash בדומיין כדי לקבל גישת Administrator לשרת. אז הוא שלח את קובץ הסיסמאות של האתר למתקיף (exfiltration).
(זו שאלה פתוחה ויש וריאציות רבות, כל סיפור עשיר ואמין התקבל).

שאלה 6 (10 נק')

תנו תיאור קצר ומדויק (2-3 משפטים) של ההתקפות הבאות:

א. [5 נק'] Cache side-channel attack

התקפת גניבת מידע מקומית בין תהליכים או מכונות וירטואליות, המנצלת תחרות על משאב משותף (זיכרון מטמון). המתקיף מואט על ידי שימוש במטמון של הקורבן, וכך מסיק לאיזה כתובות זיכרון הקורבן ניגש, ומכאן מסיק (לדוגמה) מפתחות הצפנה.

ב. [5 נק'] DNS cache poisoning attack

התקפת רשת על שרתי DNS מקומיים (caching server) או תחנות קצה, הגורמת לתשובות שקריות לשאילתות DNS - לרוב, כדי להסיט גישה לדומיין ידוע אל כתובת IP של המתקיף. שיטה אחת היא להציף את הקורבן בתשובות לשאילתות אפשריות, בתקווה שמי מהן תתאים לשאילתה שהקורבן שלח ותגיע לפני התשובה האמיתית.