



תכנית לימודים להתמחות

הוראת סייבר

Cyber & OS

גל בר-און	כתיבה ועריכה
חגית כהן	
ד"ר אבי כהן	ייעוץ, הכוונה ופיקוח
פרופ' שלומי דולב	

תקציר התוכנית

תכנית הלימודים בתחום הסייבר נועדה לספק לתלמידים ולתלמידים חשיפה ראשונה ובסיסית לעולמות הסייבר, ההופכים לאחד ההיבטים הקריטיים ביותר בחיי היום-יום של הפרט, הארגון והמדינה. התחום עבר קפיצה אדירה הן ביכולות והן בסיקור התקשורתי.

מרחב הסייבר מתייחס כיום למרבית ממרחבי החיים – החל מרשתות החשמל, המים והרמזורים, דרך מערכי המחשוב הכוללים מחשבים טלפונים וכל התקן דיגיטלי, ועד לעולם ה-IOT הכולל כיום חיבור לאינטרנט של כמעט כל מכשיר או כלי כגון: רכבים, מזגנים, ביגוד, שערי כניסה ועוד.

ההגנה על מרחב זה דורשת מערך של פתרונות חדשניים וטכנולוגיות צופות פני עתיד, להתמודדות עם חולשות ואיומים משתנים ומתפתחים. מרחב סייבר מוגן ובטוח הוא תנאי הכרחי לקיומה של שגרה במאה ה-21.

אבטחת מרחב הסייבר מתייחסת לדרך האסטרטגית והטכנולוגית להגן על מערכות, רשתות ותוכניות מפני התקפות דיגיטליות. מטרות התקפות סייבר הן בדרך כלל לייצר גישה, שינוי או השמדה במערכות המחזיקות מידע רגיש. כמו כן, סחיטת כסף ממשתמשים או הפרעה לתהליכים עסקיים או לאומיים.

יישום מנגנוני אבטחת סייבר אפקטיביים ויעילים הינו תחום מאתגר, במיוחד בעולמנו היום, עולם שמכיל מרחב דיגיטלי עצום שחשוף להתקפות סייבר חדישות ומתוחכמות.

חובה להדגיש כי תחום הגנת הסייבר, מעבר לצורך המובן בשימוש בטכנולוגיה, מחויב לעמוד בכללי האתיקה ועל כן יש לשלבם בתכנית הלימודים ולהדגיש את האחריות הכרוכה בלימוד התחום. גם בפרק העוסק במנגנוני הגנה כנד קוד זדוני קיימת הקפדה על שימוש באמצעים אשר אינם גורמים נזק ממשי למחשב או למערכותיו השונות.

מטרת התוכנית

מטרת העל של התוכנית היא לייצר אצל הלומדים בסיס ידע ראשוני אך משמעותי להבנת התחום, חשיבותו, ואופק ההתפתחות של האיומים והחולשות. הבנה זו תאפשר להמשיך ולהתמקצע, מתוך ראיית רוחב ושמירה על כללי אתיקה אשר ילוו את התלמידים בהמשך חייהם המקצועיים והאישיים.

הכשרת תלמידי תיכון במגמת הנדסת תוכנה ברמה אנליטית גבוהה בתחום הגנת מידע במרחב הסייבר הכולל זיהוי תקיפות ואנומליות על רשתות ופרוטוקולי תקשורת, שיטות קריפטוגרפיות עליהן מבוססת אבטחת האינטרנט, זיהוי קוד זדוני, אבטחת רשתות תקשורת ומנגנוני מערכות הפעלה.

הלימודים עונים על צורך ברור להכשרת תלמידים מצוינים בתחום אבטחת המידע והסייבר ומעניקה הזדמנויות מעשיות להשתלבות בתעשיות עתירות ידע. לימודי התוכנית מספקות ידע רב והכשרה מצוינת המתאימה לסביבת עבודה טכנולוגית

המתמחה באבטחת מרחב הסייבר.

הדרישה בתעשייה לבוגרים בנושא הגנת מערכות סייבר הינה מהגבוהות בכל התחומים.

פרקי לימוד בהתמחות

שם הנושא	שעות
פרק 1 - מבוא להגנת סייבר	10
פרק 2 - הרחבת הידע בשפת התכנות	25
פרק 3 - תכנות מתקדם	25
פרק 4 - היכרות ראשונית עם מודל 5 השכבות	20
פרק 5 – פרוטוקולים במודל 5 השכבות	20
פרק 6 - תכנות תשתיות רשת מעל פרוטוקולי תקשורת	40
פרק 7 - הגנת סייבר בסביבות מבוססות רשת	45
פרק 8 - קריפטוגרפיה	40
פרק 9 - אבטחת מערכות הפעלה	45
פרק 10 - הגנת סייבר מפני קוד לא ידוע	50
פרק העשרה - שיטות לזיהוי והגנה מפני תקיפות סייבר בסביבת WEB	
פרק 11 – רשימת נושאים עבודות גמר לדוגמא	

המלצה: להתחיל את הלימוד עם פרקים 6 – 1 ולאחר מכן ניתן להתמקד בשניים מתוך פרקים 7 – 10.

פרק 1 – מבוא להגנת סייבר

הקדמה:

פרק זה מהווה כניסה לעולם הסייבר. התלמידים נחשפים לעולם הסייבר, לסכנות ואיומים הקיימים והפתרונות לאיומים אלו.

מטרות כלליות:

סקירת עולם הסייבר – הבנת מגוון התקני הקצה החשופים לחולשות, וסוגי התשתיות של ארגונים – פנימיות, בענן, ובעיקר היברידיות.

סקירת IOT והבנת משמעויות הסייבר בעולם הנגזרות מעולם ה IOT
דוגמאות למערכים עליהם יש להגן - לאומיות מסחריות ואישיות – חברת חשמל, הבורסה, ואפילו המקרר הביתי

היכרות עם עולם הסייבר תוך התמקדות בהגנת סייבר.

הכרת מושגי יסוד כגון: סייבר, חולשות ואיומים, סוגי הגנות ועוד.

מטרות ביצועיות:

התלמיד יסביר את הצורך והאיומים הקיימים המצריכים הגנת סייבר.

התלמיד יסביר את השימוש בהגנת סייבר כדרך התמודדות עם האיומים.

התלמיד יבחין ויסביר מהם הישויות המרכיבות את העולם המכונה "סייבר" – התקני קצה (מחשבים, טלפונים, רכבים, שעונים מקררים..), רשת, אינטרנט.

התלמיד ינתח ויתאר אלגוריתם נתון.

מושגים והכוונה:

רשת אינטרנט

עולם הסייבר

התקפת סייבר

הגנת סייבר

דרכי הוראה:

פרק זה הינו פרק עיוני ויילמד ע"י ניתוח משימת קיץ.

דרכי הערכה:

עבודה מעשית בסיסית בתחום הסייבר.

חלוקת השעות:

שעות	נושא
2	אתיקה
1	איומי עולם הסייבר
2	היכרות עם עולם הסייבר תוך התמקדות בהגנת סייבר
2	הישויות המרכיבות את העולם המכונה "סייבר"
3	ניתוח אלגוריתם נתון
10	סה"כ שעות:

פרק 2 - היכרות חוזרת עם שפת תכנות

מטרות כלליות:

הכרת סביבת העבודה.

העמקת ההיכרות עם שפת תכנות כדוגמת: פייתון.

*ניתן להשתמש בכל שפת תכנות אחרת.

מטרות ביצועיות:

התלמיד יתקין את סביבת העבודה ויריץ תוכנית בסביבה זו.

התלמיד יפתח בגישה מודולרית פתרון אלגוריתמי הכולל שימוש בהוראות תנאי ולולאות.

התלמיד יפתח בגישה מודולרית פתרון אלגוריתמי הכולל הגדרת פונקציות (מחזירות ושאינן מחזירות ערך).

התלמיד יפתח בגישה מודולרית פתרון אלגוריתמי המשתמש במבנה נתונים.

התלמיד יבחין בין טיפוסים משתנים שונים.

התלמיד ידע להגדיר מילון לאתחל אותו ולאחזר נתונים ממנו

התלמיד יממש פתרון המצריך ייבוא ושימוש בספריות חיצוניות בשפה כגון: Random,

מושגים והכוונה:

טיפוס נתונים בוליאני.

תנאים בוליאניים פשוטים ומורכבים.

הוראות תנאי וביצוע מותנה.

בדיקת תקינות קלט.

ייבוא ושימוש בספריות בשפה.

מחרוזות כטיפוס נתונים מורכב.

פונקציות מקבלות / אינן מקבלות פרמטרים.

ערך מוחזר מפונקציה.

תחום הכרה של משתנים,

לולאות – מספר חזרות ידוע מראש או מותנה בתנאי לוגי.

מחרוזות, פעולות על מחרוזות.

מבנה נתונים שונים: מערך, רשימה, Tuple, מילון

רשימה כמבנה נתונים דינמי, פעולות על רשימה

דרכי הוראה:

הפרק יילמד תוך ביצוע משימות מובנות ומתמשכות לרמה של תוכנית מורכבת.

ניתן להיעזר בקורסים מקוונים כדוגמת self.py ו next.py של המרכז להוראת הסייבר או קורס תכנות של אוניברסיטת ת"א או כל אוניברסיטה אחרת.

דרכי הערכה:

בחינה מעשית הכוללת פיתוח ומימוש של פתרון אלגוריתמי עבור משימה מורכבת.

חלוקת השעות:

שעות	נושא
2	סביבת עבודה
2	משתנים
2	הוראות תנאי
2	ביצוע חוזר
4	מחרוזות ופעולות על מחרוזות
3	פונקציות
2	שימוש בספריות
8	מבנה נתונים – רשימה, מערך, מילון, Tuple
25	סה"כ שעות:

פרק 3 – תכנות מתקדם

מטרות כלליות:

הפרק הינו המשך ליחידה הרביעית במדעי המחשב ומטרתו הקניית עקרונות מתקדמים בתכנות מונחה עצמים תוך התנסות בבניית מחלקות. הכרת מושגי יסוד של תכנות מונחה עצמים מתקדם, הורשה, ממשקים, מחלקה מופשטת, המרה כלפי מעלה וכלפי מטה, דריסת פעולות, זימון פולימורפי. הכרת מבני נתונים נוספים בשפה, ספריות מרכזיות ושימושיות בשפה.

מטרות ביצועיות:

העמקת ההיכרות עם שפת התכנות. הבנה ויכולת תכנון וכתיבה מתקדמת של תכניות מחשב. מימוש ויישום סקריפטים הכוללים את הידע הנדרש לתוכנית. הטמעת מבני הנתונים ילמדו בהדרגה על סמך ההתקדמות בחומר הבגרות. כתיבת קטעי קוד הכוללים חלוקה לפונקציות. הטמעת תכנות מונחה עצמים. כתיבת תכניות המשתמשים בספריות (modules) מרכזיים ונפוצים בפיתוח. תיבת סקריפטים הכוללים חלוקה פונקציות. כתיבת קוד מתועד על פי פורמט PEP8. כתיבת קטעי קוד הכוללים ספריות עיקריות בפיתוח, כדוגמת: os, sys, pillow, shutil, math, subprocess, urllib, glob, flask

מושגים והכוונה:

טכניקות איטרטיביות באמצעות Generator
אובייקט דינמי תוך שימוש ב- Decorator
מבוא לטכנולוגיה הנבחרת כגון: ניתוח מידע, בניית ממשק גרפי, עיבוד תמונה, ניתוח ועיבוד קבצים, streaming
טיפוסים ומבני נתונים
ספריות ויישומם בקוד
עיבוד מידע
תכנון ה output
הצגת מידע גרפית, טקסטואלית או קובץ.

דרכי הוראה:

הפרק יילמד תוך ביצוע משימות מובנות ומתמשכות לרמה של תוכנית מורכבת ממספר נושאים. המורה יבחר להתמקד בנושא מרכזי לבחירתו כגון: ניתוח מידע, בניית ממשק גרפי, עיבוד תמונה, ניתוח ועיבוד קבצים, streaming ועוד. התלמיד יחקור את המודולים המתאימים לנושא הנבחר.

דרכי הערכה:

בחינה מעשית הכוללת פיתוח ומימוש תוכנית המתבססת על המודולים שנלמדו. התוכנית יכולה להיות מוצגת כתרגיל או אוסף תרגילים "מתגלגלים" המובילים ליישום סופי שלם.

חלוקת השעות:

שעות	נושא
5	מבני נתונים
3	תכנות מונחה עצמים
3	עבודה מול ספריות מתקדמות
5	ספריות ויישומם בקוד
9	עיבוד מידע, streaming, מבוא לעיבוד תמונה
25	סה"כ שעות:

דוגמאות לתכנים:

פיתוח ממשקים בסביבת TKinter, יצירת היסטוגרמות והצגתן, יצירת מערכים עם ספריית numPy תוך מתן אינדקס, ניתוח מידע בסיסי עם ספריית panda, הצגת תמונות באמצעות ספריית openCV, עבודה מול קבצים ווידאו, object detection, video streaming, עיבוד תמונה (contour, grid, edge).

עיבוד תמונה דרך numPy ו openCV

התקנת סביבת עבודה כדוגמת: anaconda ו notebook

לימוד פונקציות מרכזיות ב NumPy and Image Basics

לימוד Image Basics with OpenCV

עיבוד תמונה באמצעות ספריית OpenCV

לימוד מספר פונקציות עבור זיהוי תמונה - Object Detection
ניתן להרחיב את הידע לעבודה עם קבצי וידאו Object Tracking

פרק 4 – היכרות ראשונית עם מודל 5 השכבות

הקדמה:

בפרק זה נחשוף את התלמידים למודל 5 השכבות. חשוב להדגיש את התלויות ועקרון ההכמסה בין השכבות השונות.

* בפרק הבא התלמידים ילמדו פרוטוקולים שונים תוך מתן דגש לנבכי הפרוטוקולים, מרכיביהם ומבנה כל פרוטוקול.

מטרות כלליות:

הפרק עוסק במודל 5 השכבות: פיזית, קו, רשת, תעבורה ואפליקציה. מומלץ להציג את הרעיון הכללי בחלוקה לשכבות, מטרות כל שכבה ודרכי הפתרונות לבעיות דרך הפרוטוקולים.

היכרות עם כלי ניתוח התוכנית – debug.

היכרות עם כלי ניטור מידע ברשת – sniffer.

מטרות ביצועיות:

התלמיד יסביר את מודל השכבות.

התלמיד יסביר כל שכבה במודל, תפקידה וחשיבותה.

התלמיד יסביר את עקרון ההכמסה במודל.

התלמיד יתנסה בשימוש בכלי בדיקת תוכנה – debug.

התלמיד יתנסה לראשונה בכלי צפייה במידע – sniffer.

מושגים והכוונה:

השכבה הפיזית – הכירות בסיסית על תהליך איפנון המידע הדיגיטלי על גל אנלוגי

שכבת הקו – הבנת תפקידיה של שכבת הקו ומיקומה במודל השכבות.

שכבת הרשת – הבנת תפקידיה של שכבת הרשת ומיקומה במודל השכבות.

שכבת התעבורה – הבנת תפקידיה של שכבת התעבורה ומיקומה במודל השכבות.

שכבת האפליקציה – הבנת תפקידיה של שכבת האפליקציה.

כלי debug – מעקב אחרי קוד.

ניתור מידע ב – sniffer.

דרכי הוראה:

פרק זה הינו תיאורטי ומהווה בסיס ידע לפרק הבא. הפרק יילמד תוך הסבר כל שכבה, תפקידה ומיקומה במודל 5 השכבות. כמו כן יש לתת דגש על התלויות ועקרון ההכמסה בין השכבות השונות.

התלמיד ייחשף לראשונה לכלי מעקב אחרי קוד (debug) וניתור מידע ברשת (sniffer).

דרכי הערכה:

בחינה עיונית הכוללת תיאור זרימת מידע ברשת מרגע שליחתו ועד לרגע קבלתו ביעד.

חלוקת השעות:

שעות	נושא
2	תיאורית מודל 5 השכבות
5	הסבר מעמיק על תפקידי כל שכבה
2	עקרון ההכמסה
2	מעקב אחרי קוד (debug)
3	ניתור מידע ברשת (sniffer)
2	בחינה.
15	סה"כ שעות:

פרק 5 – פרוטוקולים במודל 5 השכבות

הקדמה:

פרק זה מהווה המשך לפרק הקודם. התלמידים למדו את מודל 5 השכבות וכעת ילמדו פרוטוקולים שונים במודל וכלים שימושיים.

מטרות כלליות:

בפרק זה נסקור ונכיר פרוטוקולי תקשורת מרכזיים ונשלב תכנות פרוטוקולי תקשורת. הפרק מורכב מחלק תיאורטי וחלק מעשי ותכנותי. התלמידים ילמדו פרוטוקולים בתקשורת ואת המוטיבציה לשימוש בהם ככלים לפתירת תקלות.

מטרות ביצועיות:

התלמיד יתנסה בכלי ניתור מידע – sniffer ללמידת פרוטוקולי תקשורת וביצוע מחקרים. התלמיד יתאר את ההבדל בין צפייה במידע באמצעות sniffer אל מול קבלתו. התלמיד יסביר כיצד ניתן לנתח סקריפטים העברת נתונים ותקשורת בין שרת ולקוח. התלמיד יסביר כיצד לנתח קובץ pcap - Wireshark. התלמיד יבחין בין הפרוטוקולים השונים והשכבות בתוכו ויבצע filtering למציאת מידע עבור כל שכבה (לדוגמה host sender name, ip src host port). התלמיד יסביר כיצד ניתן להסיק מסקנות מתוך קבצי pcap את ה metadata כגון: גרסת השרת, גודל המידע, גרסאות הדפדפן, GET/POST. התלמיד יממש סקריפטים המדמים פקודות CMD כגון: ping, netstat, nslookup. Traceroute. התלמיד ינתר את תעבורת הנתונים שיצר באמצעות Wireshark וינתח את הגעתן לידע באמצעות תוכנות קצה מתאימות כגון דפדפן. התלמיד ישלח פקטות HTTP ב Get/Post, ינתח את ההבדלים בין request ו response דרך chrome debugger. התלמיד יסביר כיצד לחשב Subnet. Mask, את הקשר לפרוטוקול DHCP והודעות השונות. התלמיד יסביר כיצד להבין את כלל השדות ב Headers של הפרוטוקולים HTTP ו DNS. התלמיד יידע להסביר מהם handshake, connection, sessions, checksum, keepalive, sliding window, sequence number לאיזה פרוטוקול הם שייכים ומה שימושם. התלמיד יידע להסביר את הבדלים בייצוג הנתונים בין פרוטוקול IP, DNS, TCP, UDP, HTTP, ARP.

מושגים הכוונה:

רכיבי Hub

Switch, Frame, MAC, Ethernet

המושגים: Subnet Mask , Broadcast , Loopback

פרוטוקולים Ethernet ו-ARP.

רכיב Router ופרוטוקולים DHCP, ICMP, IP

המושגים: route TTL, MTU, NAT

המושג port והפרוטוקולים TCP ו UDP.

המושגים: Netstat , Ports, Checksum, Sequence Number

פרוטוקולים בשכבת האפליקציה: DNS, HTTP, TLD

פרוטוקולים Zone Transfer, SMTP

המושגים: chrome debugger ו- sniffer Wireshark

דרכי ההוראה:

יש להציג לתלמידים את הרעיון העומד בבסיס מודל השכבות ולהדגיש את הקשר בין השכבות.

להציג את המידע והשירות שכל שכבה מספקת לשכבה שמעליה /מתחתיה.

ללמד את המאפיינים של הפרוטוקול באמצעות Wireshark

ללמד את מבנה הפרוטוקול (headers). מומלץ להדגים את המבנה באמצעות Wireshark. יש

לשים דגש עם שדות חובה/אופציונליים

דרכי הערכה:

כתיבת פילטרים בכל שכבה. הצגת אלגוריתם נפוץ אחד מכל שכבה. ניתן להוסיף בחינה עיונית.

חלוקת השעות:

שעות	נושא
3	בניית פילטרים וניתור מידע – sniffer
4	ניתוח קובץ pcap - Wireshark
3	כתיבת סקריפטים פקודות CMD
3	ניתוח תעבורה משולבת sniffer ודפדפן
3	ניתוח תעבורה דרך chrome debugger

3	ניתוח header ומאפיינים עיקריים בכל שכבה
3	ייצוג הנתונים בין והבנת ערכיהם בפרוטוקולים מרכזיים
2	זרימת המידע בין הפרוטוקולים
2	בחינה/עבודה מסכמת
25	סה"כ שעות:

הסברים ודוגמאות ניתן לראות בנספחים תחת הכותרת: הרחבה לפרק פרוטוקולים במודל 5 השכבות.

פרק 6 – תכנות תשתיות רשת מעל פרוטוקולי תקשורת

הקדמה:

בפרק זה הינו מעשי תכנותי.

התלמידים יישמו את החומר הנלמד בפרק הקודם הנוגע לתקשורת נתונים, יכתבו תכניות לרכיבי התקשורת תוך שימוש ויישום הפרוטוקולים שנלמדו.

מטרות כלליות:

בפרק זה נשלב תכנות פרוטוקולי תקשורת.

תכנות sockets - פיתוח תוכניות בסביבת הרשת באמצעות socket-ים

תכנות sockets מתקדם: ריבוי משתמשים

תכנות רכיבי תקשורת כתוכנה וירטואלית : switch, hub ועוד

ניתוח וייצור תעבורת רשת בעזרת חבילת scapy של Python.

בנייה ושליחת פקטות (חבילות מידע) עבור פרוטוקולים: TCP, IP, ARP, Ether, DNS, ICMP

יישום ומימוש sockets מעל שכבת התעבורה: מימוש Three Way Handshake

מימוש sockets מעל UDP

פיתוח ויישום תכנית מעל שכבת אפליקציה client-server פיתוח מעל פרוטוקול SMTP,

בפקודות השונות הקיימות בפרוטוקול HTTP

מטרות ביצועיות:

התלמיד יסביר במילים שלו את מבנה ה socket ומאפייניו.

התלמיד יישם פרוטוקול TCP כשרת לקוח באמצעות socket. יש לשים דגש על:

1. Communication Breakdown

2. Viewing Socket State

3. Closing Connections

4. Blocking Calls

התלמיד יפתח sockets בסביבת multithreads

התלמיד ינהל מספר connections (חיבורים) במקביל

התלמיד ינתר את תעבורת הנתונים ביישומים שפיתח (באמצעות wireshark)

התלמיד יסביר במילים שלו את מבנה הפקטה בכל פרוטוקול, יבנה ישלח ויקבל פקטות

בפרוטוקולים שונים דרך מודול scapy

התלמיד יישם אלגוריתם של ניתוח המידע בהתאמה לפרוטוקולים שנלמדו בפרק הקודם

התלמיד יפתח מימושים מתקדמים דרך sockets ו scapy כדוגמת שרתי HTTP, Peer-To-

Peer, SMTP

דגשים מקצועיים ופדגוגיים:

מומלץ לתת לתלמידים לפתח יישום הנוגע בכל אחת מהשכבות שנלמדו תוך כדי מהלך הלימוד של כל שכבה. פרק זה שם דגש חזק על: תכנות דרך sockets, פיתוח שרת מרובה תהליכים, יישום מבנה הפרוטוקולים (צד לקוח/שרת), יצירת פקטות וניתוחם.

דרכי ההוראה:

להתחיל עם הסבר על מבנה socket ומאפיניו, פיתוח שרת ECHO פשוט, הוספת פקודות מתקדמות בשרת, כגון PIL, pypclip, screenshot ועוד. פיתוח פרוטוקול עצמאי מעל TCP ולבדוק גדלי החוצצים (buffers) בשרת ובלקוח לימוד מאפיינים מתקדמים של timeout ו bing backlog HTTP בסביבת request ו regex באמצעות מודול על מנת ליישם את בניית וניתוח header בפרוטוקול. לימוד בניית פקטות באמצעות scrapy מלמטה ללמעלה. לנתר את שליחתן באמצעות Wireshark

יש לשים דגש על מאפייני הפרוטוקולים כדוגמת: HTTP GET/POST, TCP, לעשות בדיקות ולידציה בשרת ובלקוח עבור מספר חיבורים במקביל וגודל הודעות מקסימלי. Arp request/response/broadcast ICMP - חותמת זמן, בקשת מידע, תשובת מידע, בעיית פרמטר, תשובת מסכת ניתן לתרגל יישום פרוטוקול אחד באמצעות פרוטוקול אחר. לדוגמה: ליישם פרוטוקול TCP באמצעות PING (שהרי ping מסוגל לשמור מידע ב payload). במקרה זה, מטרת התלמידים תהיה ליישם אימות (validation) של פרוטוקול TCP על גבי PING.

דרכי הערכה:

התלמידים יכתבו עבודה מעשית המתייחסת לנושאים המפורטים בדרכי ההוראה.

חלוקת השעות:

שעות	נושא
4	תכנות sockets בסיסי
4	תכנות מבוסס threads

5	תכנות sockets מתקדם מרובה משתמשים
5	בניית שרת מתקדם הכולל שימוש בספריות
4	פיתוח פרוטוקול עצמאי מעל TCP
4	פיתוח שרת בסביבת HTTP
3	פקטות באמצעות scapy
4	בדיקות ולידציה בשרת ובלקוח
5	פיתוח ובניית פרוטוקול עצמאי
2	הערכה – הגנה על עבודה/מבחן
40	סה"כ שעות:

הסברים ודוגמאות ניתן לראות בנספחים תחת הכותרת: הרחבה לפרק תכנות תשתיות רשת מעל פרוטוקולי תקשורת.

פרק 7 - הגנת סייבר בסביבות מבוססות רשת

הקדמה:

פרק זה עוסק בסוגי החולשות ואיומים הקיימים הן בסביבת התשתית והן בסביבת התקן הקצה. סוג ההגנה הינו שונה בכל אחד מהרכיבים. ההקדמה תכלול סקירת סוגי ההגנות בכל היבט ובהמשך הפרק המיקוד הינו בהתקן קצה. יש להדגיש את חשיבות האתיקה וההגבלות המשפטיות הכרוכות בהקפדה על חוקי הסייבר.

אתיקה:

רציונל הקו המנחה את הוא להציג לתלמידים את האזורים שהם "שחורים" - כלומר מהווים עבירה פלילית. השאיפה לצמצם את האזור ה"אפור" ככל שניתן, ולספק לתלמידים כלי בסיסי כדי להבין אם פעולה מסוימת היא לגיטימית או לא - אמנת אתיקה בסייבר. אנו ממליצים בחום להחתים את התלמידים, ביחד עם הוריהם והסגל (מורה ועוזר הוראה) על אמנת אתיקה בסייבר. החתימה נותנת משנה תוקף לנושא, מייחדת אותו ומדגישה את חשיבותו.

מטרות כלליות:

הכרת חולשות נפוצות בשכבות השונות ואמצעים להתמודדות מולן.
חסימת הורדה של קבצים אשר מכילים בשם הקובץ אחת מהסיומות הנפוצות.
הכרת טכניקות הגנה למניעת איומים מתקדמים כנגד פרוטוקולים נבחרים.
בעיות אבטחה נפוצות.
חורים במנגנוני אבטחה לוגיים / תוכנותיים ודרכי התמודדות.

מטרות ביצועיות:

התלמיד יכיר לפחות שתי חולשות אפשריות מהשכבות השונות. יש להתמקד בהבנה של החולשה וכיצד ניתן לבנות מערכת הגנה אבל חולשה זו.
התלמיד ייפתח מערכת הגנה כנגד החולשה שנסקרה.
התלמיד יישם לפחות 2 מנגנוני הגנה.
* רשימת המנגנונים בפרק נספחים תחת הכותרת הרחבה לפרק הגנת סייבר בסביבות מבוססות רשת - מנגנוני הגנה.

מושגים והכוונה:

חולשות ואיומים בשכבות השונות.
טכניקות הגנה.

דגשים מקצועיים ופדגוגיים:

מטרת נושא הלימוד הוא הקניית יכולת מחקר וניתוח רמת האבטחה של פרוטוקולי תקשורת שונים בקרב התלמידים בנושא. במהלך הנושא יוצגו דוגמאות להגנות על איומים כנגד פרוטוקולים שונים ואמצעי הגנה להתמודדות מולן. התלמידים יתנסו בעבודות מעשיות של איתור חורי אבטחה פוטנציאליים בפרוטוקולי תקשורת ופיתוח הגנות אל מול איומים אלו.

דרכי הוראה:

מומלץ להתחיל עם תוכנה מסוימת עבור הדגמת חולשה בפרוטוקול הנבחר. למשל ניתן להשתמש ב clian&able או Kali-Linux על מנת להבין את חולשת מנגנון ARP. מומלץ להקפיד על עקרון מטרת הלימוד שהוא לבנות מערכת הגנה עבור החולשה שנלמדה כך שאין להשאיר חולשה ללא הגנה. כחלק מבניית מערכות ההגנה כדוגמת VPN ו TLS ניתן לשלב הצפנה סימטרית וא-סימטרית, אבל רק ברמת API מובנים (עדיין לא להיכנס לתחום הקריפטוגרפי). הדגמת חולשות הפרוטוקולים ברמה של חורי אבטחה ולהדגיש במדויק תוך מחקר מצד התלמידים היכן מתקיים חור האבטחה. לאחר חקר ומציאת הבעיה מומלץ להעלות סוגיות ורעיונות כיצד ניתן לאבטח מפני איום אפשרי. מומלץ לתרגל את טכניקת spooning ולתת לתלמידים ליישם TCP Port Knocking להתמודדות עם איום זה. אסטרטגיית הוראה אפשרית היא לתת לתלמידים להשלים את החלקים הרלוונטיים (פונקציות) לתת הפרק (התקפה/הגנה), מאשר לכתוב את כל מכלול התוכנית.

דרכי הערכה:

בפרק זה על התלמיד להבין לפחות איום אחד ולכתוב הגנה מפניה. מומלץ לספק סביבה וירטואלית החשופה למתקפה ולתת לתלמידים לכתוב את אלגוריתם ההגנה מפניה. לדוגמה ניתן להרים שרת HTTP שעליו אפליקציית web החשופה למתקפת SQL Injection. על התלמידים למצוא את חור האבטחה, ולכתוב שכבת הגנה.

חלוקת השעות:

שעות	נושא
2	אתיקה בסייבר
6	הדגמת חולשות הפרוטוקולים ברמה של חורי אבטחה

6	מנגנוני הגנה כנגד החולשות שנסקרו
8	חורים במנגנוני אבטחה לוגיים/תכנותיים
10	בניית מערכות ההגנה עבור חור אבטחה קיים
10	חקר עבור הבעיה אבטחה והתמודדות כנגדה
3	הערכה – הגנה על עבודה/מבחן
45	סה"כ שעות:

* הסברים ודוגמאות ניתן לראות בנספחים תחת הכותרת: הרחבה לפרק הגנת סייבר בסביבות מבוססות רשת.

פרק 8 - קריפטוגרפיה

הקדמה:

קריפטוגרפיה מודרנית מספקת אלגוריתמים ופרוטוקולים להגנת ישויות המנסות לבנות אמון או שמספקת הגנה כנגד גורמים זדוניים שיכולים לצותת לתקשורת או לשנות אותה. נושאים בסיסיים בתחום הקריפטוגרפיה כוללים הצפנה מאובטחת, חתימות דיגיטליות, ואימות. בפרק זה נדון בנושאים אלו, במימוש שלהם וניישם אותם דרך תכנות. הנושא ישלב בנוסף, רקע מתחום תורת המספרים וזאת על מנת להבין כיצד מערכות הצפנה מודרניות עובדות כגון RSA.

מטרת כלליות:

פרק זה עוסק בתורת ההצפנה – קריפטוגרפיה. התלמידים ייחשפו לסוגי הצפנות שונות. היכרות עם מערכות הצפנה קלאסית ומערכות הצפנה מושלמת.

מבוא להצפנה סימטרית OTP תוך יצירת מפתח רנדומלי באמצעות PRNG / פונקציית HASH

קריפטוגרפיה סימטרית: DES

בחירת אחד מהנושאים הבאים:

- חתימה דיגיטלית באמצעות אלגוריתם Lamport.
- קריפטוגרפיה ציבורית, מערכות קריפטוגרפיות וחתימה דיגיטלית באמצעות RSA.

נושאי העשרה:

קריפטוגרפיה סימטרית: AES

יצירת מפתח סימטרי "באוור" באמצעות merkle puzzle

יצירת PKI מפתח סימטרי על ידי RSA בעזרת Certificate Authority ומעבר ל AES.

מטרות ביצועיות:

התלמיד יסביר וייצור הצפנה באמצעות אלגוריתם OTP ו- PRNG.

התלמיד יסביר וייצור מפתח סימטרי "באוור" באמצעות merkle puzzle

התלמיד יסביר מהי הצפנה סימטרית באמצעות DES.

התלמיד יסביר חתימות דיגיטליות.

התלמיד יסביר וייצור מערכות חתימה דיגיטלית באמצעות Lamport one-time signature

scheme באמצעות פונקציית גיבוב (hash)

התלמיד יסביר מערכות קריפטוגרפיות RSA.

התלמיד יסביר את ההבדל בין מערכת קריפטוגרפית לא מושלמת למערכת מושלמת.

התלמיד יישם בניה של מערכת סימטרית בעלת מפתח יחיד.

התלמיד יסביר במילים שלו מהי חתימה חוקית על מסמך, כיצד ניתן לייצר התקפה על מערכת חתימה דיגיטלית (methods) וכיצד ניתן לחזק את תהליך האישור (אוטנטיקציה).

בחירה: התלמיד יישם מערכת קריפטוגרפית ציבורית דרך מערכת RSA.

בחירה: התלמיד ירחיב את בניית מערכת החתימות הדיגיטליות דרך יישום אלגוריתם RSA.

דרכי הוראה:

הצגת מערכות הצפנה פשוטות כדוגמת צופן קיסר. מימוש אלגוריתם המיישם הצפנת קיסר ומתקפת BF הכוללת prefix ו suffix של ההודעה למציאת מפתח הסיבוב. הצגת אלגוריתם סיבוב נוסף כדוגמת Transposition. התלמיד יכתוב את אלגוריתם ההצפנה, פיענוח ומתקפה.

פיתוח אלגוריתם ליעול על מנת להפוך את OTP ליותר יעיל, במובן של שמירת מפתח קצר במקום פנקס קודים ארוך.

למידת פונקציית Hash תוך בניית אלגוריתם המגלה האם המשפט כתוב באנגלית תקינה. נלמד איך לאפשר לצדדים המעוניינים להצפין הודעות מבלי לבוא לשתף seed רנדומי ביניהם מראש אלה לייצר אותו באוויר באמצעות merkle puzzle.

תיאור אלגוריתם DES. הצגת ותכנות מערכות קריפטוגרפיות סימטריות.

לפני הצגת אלגוריתם RSA ניתן לייצר צופן פשוט יותר המתבסס על חישוב מודולרי כדוגמת Affine cipher המבוסס על הצפנה מתמטית באמצעות חשבון מודולרי.

דרכי הערכה:

עבודה מעשית תכנותית הכוללת פיתוח ומימוש של פתרון אלגוריתמי קריפטוגרפי עבור משימה מורכבת. ניתן לשלב גם בחינה עיונית המציגה בעיות קריפטוגרפיות מסדר גבוה (דוגמאות בחומרי ההוראה).

חלוקת השעות:

שעות	נושא
4	מבוא לקריפטוגרפיה ואלגוריתמים פשוטים
6	ייצור הצפנה באמצעות אלגוריתם OTP ו- PRNG
4	פונקציות HASH
6	יצירת מפתח סימטרי "באוויר" באמצעות Merkle PuzzleM

6	הצפנה סימטרית באמצעות DES
6	חתימות דיגיטליות באמצעות Lamport one-time signature
8	בניה של מערכת סימטרית בעלת מפתח יחיד.
40	סה"כ שעות:

* הסברים ודוגמאות ניתן לראות בנספחים תחת הכותרת: הרחבה לפרק קריפטוגרפיה.

פרק 9 – אבטחת מערכות הפעלה

הקדמה:

מבוא למערכות הפעלה, מבנה מערכות הפעלה, תפקידי מערכת ההפעלה. הפרק עוסק באספקטים שונים הנוגעים לאבטחת מערכות הפעלה מודרניות, החל ממודלים אבסטרקטיים וכלה במנגנונים ספציפיים במערכות הפעלה ידועות כמו: WINDOWS או UNIX. יסקרו מנגנונים בסיסיים של מערכת ההפעלה כדוגמת , Fat32 מנגנון Registry, ניהול זיכרונות ועוד. לאחר סקירת הגדרות הבסיס, יסקרו מנגנוני אבטחה המשולבים במערכות הפעלה ברמות השונות החל משכבת ה- Kernel ועד לשכבת היישומים. הטכניקות שתסקרנה כוללות ניטור, הגנה על זיכרון ותהליכים, בקרת גישה ומכונות וירטואליות. כמו כן תסקרנה שיטות לגלוי ומניעת חדירות. חלק חשוב בנושא הוא הבנת התקפות והגנות אשר מדגימים חדירה למערכת הפעלה ושיתילת קוד עוין בתוכה. דרך היישומים התלמידים ילמדו בפירוט גם את שיטות החדירה וגם את מנגנוני ההגנה על מערכות הפעלה.

מטרות כלליות:

הסבר מבנה זכרון, זכרון וירטואלי, זכרון פיזי וטבלת TLB ומנגנון MMU.
הסבר מבנה תהליך (process) והקצאתו בזכרון.
סנכרון ותקשורת בין תהליכים.
ניהול הגדרות (registry) במערכת ההפעלה.
מנגנוני אבטחת מערכות הפעלה ובקורות כניסה.
הפרדת הזכרון בין user mode ל kernel mode.
הסבר מנגנוני: trap , fork , system calls
מנגנוני הגנה ב- linux.
הגנה מפני חולשות מערכות הפעלה: BOF , shell code injection
בניית מנגנון אימות (אוטנטיקציות) על לינוקס באמצעות PAM
בידוד תהליכים ע"י טכניקת system call interposition (לדוגמא: Ospia, sys trace, prace)

רשות:

בניית מנגנון אימות AppArmor על הפצת SUSE
מנגנון ה- hypervisor וארכיטקטורת מערכת וירטואלית.
בידוד תהליכים במכונות וירטואליות.
מנגנוני chmode ו Jailkit בלינוקס.
הגנה מפני מתקפות מתקדמות על בידוד תהליכים באמצעות SSI.

מטרות ביצועיות:

התלמיד יידע להסביר את ארכיטקטורת העל של מערכת ההפעלה Windows ו/או Linux.
התלמיד יתנסה בניהול ההגדרות Windows ו/או Linux.
התלמיד יתנסה במערכת ניהול ההרשאות ב Linux. התלמיד יסביר במילים שלו את ההבדלי ניהול ההרשאות בין Linux ו Windows.
התלמיד יחקור ויסביר במילים שלו מנגנוני אבטחת המוטמעות במערכות הפעלה ובקורות כניסה.
התלמיד יסביר במילים שלו את מבנה ה kernel, תפקידיו, הרשאותיו, יצירת sys call ו traps
התלמיד יסביר במילים שלו את מבנה הזיכרון: stack, heap.
התלמיד יכיר את הרגיסטרים המרכזיים SP, EBP, ו- EIP.
התלמיד יסביר כיצד להגן מפני חולשות BOF על מערכות יעודיות.
התלמיד יכיר את מנגנון ההזדהות ב- Linux / Windows.
התלמיד יכיר את מנגנונים מגוונים לבידוד תהליכים במערכת ההפעלה.

רשות:

התלמיד יכיר את ארכיטקטורת מכונה וירטואלית ומגנון בידוד תהליכים בה.
התלמיד יסביר כיצד לבצע chmode ו Jailkit בלינוקס על מנת להגביל משתמשים למשאבי מערכת ספציפיים. מומלץ להדגיש כיצד ניתן להימלט מ jailkit
התלמיד יסביר במילים שלו את מנגנוני ההגנה למניעת פגיעה בkernel דרך מגנונים:
GSWTK, program shepherding
התלמיד יסביר במילים שלו כיצד ניתן להגן מפני פגיעה בבידוד תהליכים דרך ערוצים חשויים.

דרכי הוראה:

את הפרק יש ללמד תיאורטית תוך מתן דוגמאות.
* רשימת נושאים מפורטת ניתן למצוא בנספחים בפרק: אבטחת מערכות הפעלה.

דרכי הערכה:

2 תרגילים מעשיים:

- תרגיל ראשון בהגנה מפני BOF.
 - תרגיל שני – יישום מערכת אוטנטיקציה.
- מבחן מקיף על הנושאים העיוניים שנלמדו.

חלוקת השעות:

שעות	נושא
2	מבוא למערכות הפעלה
4	מבנה ומרכיבי מערכות הפעלה
2	מבנה תהליך (process) והקצאתו בזכרון
3	סנכרון ותקשורת בין תהליכים
2	Registry
5	מנגנוני אבטחת מערכות הפעלה ובקורות כניסה
3	הפרדת הזכרון בין user mode ל kernel mode
8	חולשות מערכות הפעלה: BOF , shell code injection
5	מנגנון ההזדהות ב- Linux / Windows
5	מגנונים לבידוד תהליכים במערכת ההפעלה
4	Jailkit בלינוקס על מנת להגביל משתמשים למשאבי
4	מנגנוני אבטחת המוטמעות במערכות הפעלה (PAM)
45	סה"כ שעות:

פרק 10 – הגנת סייבר מפני קוד לא ידוע

הקדמה:

תכניות זדוניות לוקחות חלק ברוב אירועי פריצת המחשב ואבטחה. כל תוכנה העושה דבר הגורם נזק למשתמש, מחשב או רשת יכולה להיחשב כאל תוכנה זדונית, כולל וירוסים, סוסים טרויאניים, תולעים, rootkit ותוכנות ריגול. פרק זה עוסק **בהגנת סייבר** מפני קוד לא ידוע/עוין, תוך זיהוי איתור והגנה כנגד וירוסים, תולעים ופריצות. נושא זה מתמקד בניתוח וחקר של קוד לא ידוע תוך שימוש בכלים וטכניקות לניתוח סטטי ודינמי. ניתוח תוכנות זדוניות הוא אומנות. לנתח תוכנות זדוניות כדי להבין כיצד היא עובדת, כיצד לזהות אותה ואיך להביס אותה או לחסל אותה.

מטרות כלליות:

הערה: בלמידת פרק זה נעשה שימוש בקודים שונים לצרכי דימוי אנטי וירוסים. חובה להקפיד על שימוש בקודים ואמצעים אשר אינם גורמים נזק ממשי למחשב ולמערכתיו השונים. בנושא זה נלמד ונסקור מנגנוני הגנה מפני קוד עוין לסוגיו וכיצד פועלים מנגנונים וכלים אלו לאיתור וחסימת קוד עוין. כמו כן התלמידים יחקרו ויפתחו מנגנוני הגנה מפני קוד עוין. הגנה וניתוח קוד בטכניקה סטטית. הגנה וניתוח קוד בטכניקה דינאמית בסיסית. מבנה ה-PE. טכניקות הגנה מפני הזרקת קוד.

בחירה:

הגנה מפני קוד עוין המשתמש בטכניקות הצפנה, ערפול והתחמקות קוד. ניפוי באגים וניתוח דינמי מתקדם. הגנה מפני קוד זדוני המנסה לפגוע ברמת ליבת המערכת.

מטרות ביצועיות:

התלמיד יסביר מהו קוד עוין. התלמיד ילמד טכניקות ניתוח מגוונות על מנת לאבחן קוד שעלול להיות זדוני. התלמיד יכיר כלים בסיסיים לניתוח סטטי: virus total , strings, PE viewer התלמיד יכיר טכניקת מבוא להתחמקות ואנטי התחמקות. התלמיד יכיר את החוקים הבסיסיים של ניתוח סטטי. התלמיד יכיר טכניקות נפוצות לניתוח סטטי: file ,strings, hashes, antivirus scanning .types, resources, import / export, function, packing התלמיד יסביר כיצד לזהות packing שיתבצע על PE.

התלמיד יכיר כלים בסיסים של ניתוח דינמי: process, file/registry, network
התלמיד יחקור את מתודולוגית process layout, ופונקציות בסיסיות מ WinAPI
התלמיד יתנסה בטכניקת הגנה מפני הזרקת קוד כדוגמת DLL INJECTION.

* רשימת הרחבות רשות מפורטת ניתן למצוא בנספחים בפרק: הגנת סייבר מפני קוד עוין.

דרכי הוראה:

את הפרק יש ללמד תיאורטית תוך מתן דוגמאות. מומלץ להסתכל רשימת הנושאים המפורטת
בנספחים בפרק: אבטחת מערכות הפעלה.

הכנת מעבדה - יש להתקין על כל המכונות של התלמידים virtual box ולהתקין iso של
windows 7 כמכונה וירטואלית (הסבר מפורט בנספחים).

1. קיימת חשיבות סיומת הקובץ. ניתן לעשות ניתוח סטטי (static analysis) באמצעות

VirusTotal ולפתח תוכנית החוסמת הורדת קבצים באופן סלקטיבי

2. בפרק זה מומלץ להתחיל ללמד hooking ועבודה מול ספריות WinAPI. למשל ביצוע

hook על win32 apps, winbase.h, open file function.

* רשימת נושאים מפורטת ניתן למצוא בנספחים בפרק: אבטחת מערכות הפעלה.

דרכי הערכה:

3 עבודות פרקטיות:

- עבודה ראשונה בנושא ניתוח סטטי.
 - עבודה שנייה בנושא ניתוח דנמי.
 - עבודה שלישית תכנותית להגנה מפני הזרקת קוד תוך התנסות ב WinAPI.
- כמו כן, מומלץ לבצע הגנה מעשית על עבודות אלו.

חלוקת השעות:

נושא	שעות
מבוא להגנה מפני קוד עוין	3
הגנה וניתוח קוד בטכניקה סטטית	8
הגנה וניתוח קוד בטכניקה דינאמית בסיסית	8
מבנה ה- PE	5
טכניקות הגנה מפני הזרקת קוד	9
כלים בסיסיים לניתוח סטטי ודינמי	7
ספריות WinApi	6
טכניקות hooking	6
סה"כ שעות:	50

פרק העשרה - שיטות לזיהוי והגנה מפני תקיפות סייבר בסביבת WEB

הקדמה:

פרק זה הינו פרק רשות בתחום הגנת הסייבר בסביבת web.

מטרת כלליות:

מהי חדירות ופריצה ליישומי web.

❖ היכרות תיאורטית עם חדירות לממשק קצה גרפי.

❖ חדירות לממשק קצה גרפי.

❖ הגנה מפני URI Hacking.

❖ מהם נקודות החולשה בפלטפורמת WEB

היכרות עם חולשות ביישומי web

❖ פגיעות באמצעות דפדפן.

❖ חולשת הדפדפנים בהגנה מפני הרחבות זדוניות.

HTTP Proxy

- הגנה מפני HTTP proxy.

- כלי command-line

- סריקות ו – Footprinting: Internet registrar research, DNS interrogation, General organizational research, Server discovery (ping sweeps), Network service identification (port scanning)

- Banner Grabbing באמצעות netcat

- גילוי ואיתור proxy ו firewalls באמצעות: TRACE Request, connect test , Proxy Request, TrafficShield, Netcontinuum, URLScan

טכניקות להגנה מפני פגיעה על פלטפורמות WEB

❖ הגנה מפני Point-and-Click Metasploit

❖ הגנה מפני פריצה ידנית.

❖ שיטות הגנה כגון: יישום בקרת גישה אגרסיבית לרשת, מחרוזות חיבור SQL

Cleartext, שימוש בקבצים כוללים סיומת inc.

- ❖ סריקת הרשת למציאת חורי אבטחה ופגיעות בשרתים
- ❖ IIS Hardening
- הגנה מפני התקפת הזרקת קלט
- ❖ כיצד לצפות לבלתי צפוי
- ❖ היכן למצוא ולחפש כיווני התקפת קלט
- ❖ כיצד ניתן לעקוף אימות מצד לקוח
- ❖ מנגנוני הגנה מפני התקפות הזרקת קלט נפוצות: קנוניזציה, HTML Injection, בדיקות גבולות, Command Execution
- ❖ הגנה מפני SQL Injection
- ❖ הגנה מפני XPATH Injection
- ❖ הגנה מפני LDAP Injection
- ❖ הגנה מפני Log Injection
- ❖ הגנה מפני Encoding Abuse
- הגנה מפני פגיעת אימות (Authentication) אינטרנט
- ❖ הגנה מפני איומי אימות נפוצים: Username/Password Threats
- ❖ הגנה מפני מעקפי אימות: Token Replay, Cross-site Request Forgery
- הגנה מפני פגיעת דרך הרשאת רשת (Web Authorization)
- ❖ Fingerprinting Authz
- ❖ זיהוי סריקת רשימות ACL
- ❖ זיהוי Access Tokens
- ❖ ניתוח Session Tokens
- ❖ Role Matrix
- הגנה מפני פגיעה ביישומי ניהול מבוססי Web
- ❖ ניהול שרתים מרוחק – Telnet, SSH, FTP, FrontPage
- ❖ כשל בהגדרת תצורות: הרחבות זדוניות/לא רצויות, זליגת מידע, תצורות כושלות בניהול מצבים
- הגנה מפני פגיעה ביישומי Web Clients

❖ חולשות חדירות ופגיעה ביישומי Web Client

❖ Trickery

❖ Low-privilege Browsing

❖ Firefox Security Extensions

❖ אמצעי מדידה בצד שרת

פרק 11 - רשימת נושאים עבודות גמר לדוגמא

- ❖ מנתח רשת (network analyzer) לגילוי איומי MITM על פרוטוקול ARP (inner network)
- ❖ מערכת הגנה מפני איום כופר כדוגמת ransomware. התלמיד יחקור את מנגנוני האלגוריתמים המוטמעים בקוד הזדוני על מנת לפתח מערכת הגנה ייעודית.
- ❖ פיתוח מערכת אנטי וירוס הסורקת ומנתרת נזקות וקוד זדוני. בפרויקט זה ננסה לפתח מנגנון ניתוח סטטי וניתוח דינאמי:
 - זיהוי לינקוג' סטטי
 - מיפוי טבלת ה-Export תהיה יותר גדולה מטבלת ה-Import
 - פריסת ה dll (קוד עזר עושה לעיתים שימוש ב: kernel32, user32, gdi32, ntdll)
 - זיהוי פונקציות API זדוניות כמו: URLDownloadToFile (התוכנית מורידה קובץ מהאינטרנט)
- ❖ חקר והבנה מעמיקה עבור דרך הפעולה והאלגוריתם של תוכנת keylogger המנתרת ומסננת את לחיצות המקלדת באופן סלקטיבי כפונקציה של אירועים המתרחשים בדפדפן (למשל, ה keylogger יופעל רק כאשר גולשים לאתר מסוים). התלמיד יפתח מערכת הגנה מפני התקפה זו.
- ❖ פיתוח VPN ותשתית tunneling protocol - העבודה תעסוק ביצירת רשת פרטית וירטואלית מאובטחת בין מחשב קצה שמחוץ לארגון למחשב פנימי שנמצא בתוך הרשת הפנימית (והמאובטחת) של הארגון לצורך שליחה וקבלת נתונים.
- ❖ פיתוח מערכת הגנה נגד BOF - התלמיד יבנה אמולטור המדמה stack over flow כחלק מחולשת ה BOF ויבנה עבודה מנגנון ההגנה שיכול להתבסס על אלגוריתם stack canaries או DAC
- ❖ Load Balancing HTTP Servers With Linux PAM authentication - מטרת העבודה היא לבנות מערכת של שרת HTTP ו Load Balancer (לבקרת עומסים) מבוססים על פרוטוקול הצפנה TLS שמטפל בבקשות HTTP מוצפנת מהלקוח. בנוסף, תוכנה המשתמשת בservices שמוגנת בעזרת PAM במערכת לינוקס ששולחת לשרתים הקיימים אתר חדש שיתווסף אליהם כבקשת הלקוח.
- ❖ בניית תשתית למערכת הצפנה המקומפלת כ- DLL (כך שיהיה ניתן להשתמש בה בצורה גנארית במודל שרת לקוח). הספרייה תאפשר אוסף יכולות קריפטוגרפיים שימומשו ע"י התלמיד בלבד וללא ספריות קריפטוגרפיות חיצוניות. הספרייה

תאפשר: הצפנות סימטריות (ניתן ליישם באמצעות OTP ו- PRNG, החלפת מפתחות באמצעות Markel Puzzle וחתימות דיגיטליות שניתן ליישם באמצעות lamport algorithm ועצי גיבוב).

- ❖ בניית sandbox וירטואלי המריץ קוד לא מוכר ומבצע עליו ניתוח סטטי ודינמי על מנת לזהות: הצפנת מחרוזות, Loader encryption, runtime linking, הסתרת headers legitimistic-, שינוי שנעשה ב Linking version
- ❖ כתיבת מנגנון מוניטור low-level - המנגנון מיישם הגנה כנגד חולשה במנגנון System call interposition (בידוד תהליכים). ניתן לבנות סביבה וירטואלית המיישמת container שמריץ את התהליך ה"לא מוכר" על אותה מערכת הפעלה ולכתוב מנגנון דמוי ptrace המנתר את ה- syscalls של התהליך הלא מוכר.
- ❖ עיבוד תמונה בסביבה מוצפנת – שרת המקבל stream מוצפן ממצלמה או sample images בכל אינטרוול נתון באופן מוצפן (ניתן ליישם פרוטוקול הצפנה TLS), השרת יבצע ניתוח לאובייקטים בתוך התמונה (ניתן להפעיל טכניקות ניתוח כגון: Corner Detection, Edge Detection, Grid Detection, Feature Matching) על מנת לזהות דמויות/אובייקטים מסוימים והתריע בהתאם.
- ❖ פרויקט מחקר לניתוח קוד לא מוכר באמצעות טכניקות reverse engineering ו IDA. המלצה לפרויקט מסוג זה לעבוד בשיטת "open-ended". שבהם התלמיד חופשי לבחור את שיטות הניתוח והאסטרטגיות, שבאמצעותם הוא יפרוס ויעצב את תהליך הניתוח לדוגמה, אסטרטגיה אחת היא לאתר יבוא של APIs חשודים בקוד ולמקד את הניתוח בחלקים אלה.

הרחבות לחומרי הלמידה

הרחבה לפרק פרוטוקולים במודל 5 השכבות.

בלימוד פרק זה מומלץ להציג את הרעיון הכללי בחלוקה לשכבות, מטרות כל שכבה, ודרכי הפתרונות לבעיות דרך הפרוטוקולים. לאחר מכן, מומלץ להעמיק בלמידת נבכי הפרוטוקול לפרט את מרכיביו ומבנהו.

דוגמאות לתכנים אפשריים:

הצגת פרוטוקול TCP בעיות איתן מנסה להתמודד ואת הדרכים בהן הוא עושה זאת.

❖ בעיה ראשונה – שינוע מידע רב

פתרון: פרוטוקול TCP בדומה לפרוטוקולים אחרים, מחלק את המידע הנשלח לחבילות הנקראות פקטות (packets). תפקידו של הצד המקבל הוא להרכיב אותן חזרה לכדי זרם מידע רציף.

❖ בעיה שניה – סדר ההגעה

פתרון: הצמדת מספר סידורי רץ (sequence number) (לכל פקטה ע"י הצד השולח, כך שהצד המקבל יוכל לשחזר את הסדר המקורי שלהן גם אם הן הגיעו בסדר שונה מהסדר בו נשלחו).

❖ בעיה שלישית – פקטות אבודות

פתרון: שימוש במנגנון "אישור הגעה" כאשר החבילה הגיעה ליעדה. הצד המקבל שולח חזרה את המספר הסידורי שלה, ובכך מודיע לשולח שהחבילה התקבלה, אישור זה נקרא acknowledgement או בקיצור ack.

❖ בעיה רביעית – סנכרון בין קצבי העברה

פתרון: מנגנון הנקרא "חלון הזזה" (sliding window) כל אחד מהצדדים מגדיר לעצמו את כמות המידע המירבית שהוא מסוגל לקבל (למשל מגה בייט אחד) ומודיע על כך לצד השני בכל פקטה שהוא שולח.

זה המקום להציג את אלוגוריתם GO-BACK- N

❖ בעיה חמישית – כיצד להתחיל שיחה

פתרון: "לחיצת יד תלת שלבית" (Three-way handshake) (הינו השלב הראשון בכל חיבור TCP. שמותיהן המקצועיים של שלושת ההודעות הנשלחות הם SYN, SYN-ACK ו-ACK. לחיצת יד זאת מבטיחה (בסבירות גבוהה) ששני הצדדים מוכנים לדבר אחד עם השני, ומסמלת את תחילת חיבור ה-TCP.

❖ בעיה שישית – כיצד יודעים אם יש עדיין מישור בצד השני

פתרון: פרוטוקול TCP מציע פתרון פשוט – במידה ולא נשלח או התקבל מידע במשך זמן קבוע מראש, אחד הצדדים שולח חבילת מידע ריקה שכל מטרתה היא לוודא שהצד השני עדיין שם. חבילה זו נקראת Keepalive והצד השני אמור לענות לה כמו לכל חבילת מידע אחרת. במידה והצד השני לא ענה (לאחר מספר נסיונות), ניתן להסיק שהוא כבר לא שם, ולהכריז על החיבור כמנותק.

❖ בעיה שביעית – מידע מגיע אך באופן קטוע או משובש

פתרון: פרוטוקול TCP בדומה לפרוטוקולים רבים אחרים פותר את הבעיה ע"י הוספת מספר ביקורת מיוחד לכל פקטה (בדומה לחותמת שעווה על מעטפה). המספר הזה נקרא checksum והוא תוצאה של פעולה מתמטית פשוטה על המידע הנשלח (לרוב חיבור כל זוגות הבתים).

זה המקום להרחיב על אלוגוריתם checksum

❖ בעיה שמינית – מספר חיבורים במקביל

פתרון: כאשר אנו ממענים פקטת TCP אנו לא שולחים אותה לכתובת IP בלבד, אלא מוסיפים לכתובת גם מספר הנקרא פורט (port). כאשר תוכנה מסוימת מבקשת להאזין לחיבורים נכנסים (bind), היא חייבת לציין גם את מספר הפורט אליו היא רוצה להאזין. מערכת ההפעלה דואגת שלא תהיינה שתי תוכנות המאזינות על אותו פורט. לכן, כאשר מתקבלת בקשת חיבור לפורט מסוים, מערכת ההפעלה בודקת איזו תוכנה מאזינה אליו ומעבירה את המידע אליה.

הרחבה לפרק תכנות תשתיות רשת מעל פרוטוקולי תקשורת.

פרק זה שם דגש חזק על: תכנות דרך sockets, פיתוח שרת מרובה תהליכים, יישום מבנה הפרוטוקולים (צד לקוח/שרת), יצירת פקטות וניתוחם.

דוגמאות לתכנים אפשריים:

- ❖ בניית שרת ולקוח המבוסס על פרוטוקול TCP. חשוב להדגיש עקרון הפרוטוקול דרך גודל החוצץ (buffer) אפשרי לקבל הבקשות (צד שרת) וכן יצירת header לגודל החוצץ (buffer) הרצוי עבור כל תגובה.
- ❖ צד שרת בונה header וצד לקוח בודק header לפני payload.
- ❖ שדרוג השרת מ - single threads ל multithreads. יש לשים דגש על ההבדל בין פתיחת thread עבור socket חדש, לבין המתנה על backlog.
- ❖ יצירת שרת בלבד המיישם פרוטוקול HTTP. הלקוח במקרה זה הוא הדפדפן. מומלץ להדגים מימושים שגויים של הפרוטוקול בצד השרת שיובילו לשגיאות תקשורת בצד הדפדפן ולתקנם בהתאם.
- ניתן ליישם שרת כזה דרך פירוק ה request דרך (regex) regular expression ושימוש במודולים כגון request
- ❖ יצירה ושליחת פקטות ARP ל switch ול host באמצעות Scapy. ניתן לממש כתרגיל את טבלת ה ARP למיפוי כתובות ה ip/mac של הרשת המקומית תוך התייחסות ל-subnet הנתון ואף לשחק עם הגדרות ה DHCP ולעמוד על השינויים הנגרמים בהגדרות הרשת.
- ❖ פיתוח אפליקציית crawler לניתוח מידע מעל פרוטוקול HTTP עבור אתרים נבחרים. ניתן להשתמש ב framework open source של פייתון (מודול scrapy). ניתן לייצר אלגוריתם רקורסיבי לאיסוף וניתוח מידע מתוך תיקיות בשרת מסוים.
- ❖ פה המקום לחקור את המגבלות עבור מידע מוצפן העובר דרך HTTPS.
- ❖ פיתוח אפליקציית peer-to-peer ללא שרת בין 2 לקוחות ויותר.

הרחבה לפרק הגנת סייבר בסביבות מבוססות רשת - מנגנוני הגנה.

יש ללמד שני סוגי הגנות מתוך האפשרויות הבאות:

1. פרוטוקול אבטחה ברמת Network. מומלץ ללמד את פרוטוקול IPsec והשימוש בו לבניית VPN

2. התלמיד יישם מנגנון Tunneling מוצפן בין שרת ללקוח על שכבת הרשת/תעבורה כחלק מתשתית VPN

3. פרוטוקול SSH

i. SSH עבור אבטחת רשתות תקשורת

ii. SSH Client/Server

iii. SSH בשכבת תעבורה כאמצעי לאימות מפתחות

iv. SSH בשכבת תעבורה עבור Packet Formation Exchange, ומניעה מגורמים לא מורשים להתחבר בSSH

4. הגנה כנגד התקפות נפוצות בשכבת התעבורה ואמצעי התמודדות מולן.

מומלץ ללמד עד 3 מנגנוני הגנה כנגד ההתקפות הנ"ל:

1. TCP Traffic Records

2. TCP Port Scans

3. TCP Host Sweeps

4. שימוש ב - TCP Port Knocking להתמודדות עם התקפות Spoofing ו- Replay

5. SYN Flood & TCP Hijacking Attacks

6. TCP Applications

7. UDP flood (disabled)

8. UDP Bomb

9. Chargen DoS

5. הגנה כנגד התקפות על רשת מקומית LAN – Spoofing ו Replay

a. פרוטוקול ARP

b. התקפת ARP Poisoning

c. MITM האזנה לסיסמאות עם Spoofing ARP לכל הרשת

d. ציתות והתקפת attack Replay

e. התחזות: מחשב הנמצא בין שני נתבים מתחזה בפני כל אחד מהם כנתב השני

6. הגנה מפני מתקפות על פרוטוקול HTTP והתמודדות מולן.

מומלץ ללמד עד 3 מנגנוני הגנה כנגד ההתקפות הנ"ל:

a. Semantic URL attack

b. Stored Procedures, Prepared Input Validation Attacks
Statements ומניעתה

c. SQL Injection הגנה מפני מתקפת multi-factor authentication
ומניעתה באמצעות impersonation attack

d. יישום firewall בשכבת האפליקציה לזיהוי DoS attacks

e. http session hijacking

f. שימוש ב"קנריות" להגנה מפני חולשות SSL

g. HTTPs להגנה מפני http session hijacking

הרחבה לפרק הגנת סייבר בסביבות מבוססות רשת:

התלמידים יתנסו בעבודות מעשיות של איתור חורי אבטחה פוטנציאליים בפרוטוקולי תקשורת ופיתוח הגנות אל מול איומים אלו.

דוגמאות לתכנים אפשריים:

פיתוח אלגוריתם המיישם MITM ודרכי ההתגוננות מפניו. הדוגמא מתמקדת בהתקפת ה-ARP Cache Poisoning ובדרכי ההתגוננות ממנה.

קיימות דרכים רבות לביצוע ההתקפה MITM, ואת כולן ניתן לספח לשני סוגים:

- ❖ התקפה כלפי משתמש פרטי המתבצעת בתוך רשת ה-LAN (Local area network).
היא רשת תקשורת מקומית (פנימית) המאפשרת גישה למשתמשי קצה בשטח גאוגרפי קטן. ברשת ה-LAN קיימות מספר התקפות אפשריות, ביניהן ההתקפה שעלייה ARP Cache Poisoning.
- ❖ התקפה המתבצעת מרשת פנימית לרשת גלובלית וההפך. במסגרת מתקפה זו ניתן לבצע את ההתקפה כלפי מספר אנשים במקביל. דוגמאות לרשת גלובלית הן ה-MAN וה-WAN

אלגוריתם אפשרי להתקפת ARP:

1. שליחת מנות ARP עם כתובת מקור מזויפת וכתובת היעד של מחשב מסוים ברשת המקומית.
2. כתוצאה מכך מאמין מחשב היעד שכתובת המקור המזויפת הינה תואמת את ה-Gateway המקורי וכך יוכל המחשב בעל הכתובת המזויפת להסניף פקטות העוברות בתעבורת הרשת של מחשב היעד.
3. המחשב ה"מרעיל" מנתח את הפקטות העוברות ברשת מחשב היעד כפי שמנתח הסורק את הפקטות ברשת שלו.
4. לכל מחשב יש שתי כתובות בעזרתן ניתן לתקשר עם מחשבים אחרים באינטרנט (MAC address and IP address).
5. מחשבים מתקשרים אחד עם השני באמצעות שליחת פקטות.
6. בעת שליחת פקטות ממחשב אחד לאחר הן עוברות דרך ה-switch.
7. בפקטת ARP שכבת האינטרנט בנויה מכתובת IP ידועה אך לא ידוע כתובת ה-MAC (אצל מחשב היעד - Destination).
8. על ידי שימוש ב-ARP ניתן לגלות את כתובת ה-MAC של מחשב היעד. איך זה עובד: כתובת ה-IP של המטרה היא כתובתו של הסורק. יוצרים הודעת ARP מסוג request

בהם ממלאים את כתובת ה-MAC- וה-IP של השולח ואת כתובת ה-IP של המטרה-יעד וכתובת ה-MAC- תהיה Broadcast. כל המחשבים באותה הרשת המקומית יקבלו את ההודעה והמחשב בעל כתובת ה-IP של המטרה יגיב ויחזיר הודעת ARP replay שתכיל את כתובת ה-MAC אותה רוצים לדעת. לכל מחשב שמורה טבלת ARP המכילה כתובת IP וכתובת MAC המקבילות להם, במהלך קיבולת הודעת ARP replay משתנה טבלת ה-ARP.

9. שימוש בשליחת הודעת ARP replay מזויפת על מנת להרעיל את טבלת ה-ARP של הסורק.

10. התוקף שולח הודעת ARP replay מזויפת אל מחשב הסורק כאשר כתובת IP של היעד היא כתובתו של מחשב אחר במעבדה (מאותו רשת מקומית) אך כתובת ה-MAC של היעד היא כתובת ה-MAC של התוקף.

11. כך התוקף למעשה מרעיל את טבלת ה-ARP של הסורק כך שכל המידע שיגיע את הסורק ויצא ממנו ברשת המקומית יעבור דרכו (באמצעות שימוש בהסנפת פקטות).

12. תהליך זרימת המידע ישתנה לכזה: במקום שמידע בתוך הרשת המקומית יעבור ממחשב אחד לאחר הוא יגיע קודם לתוקף והוא יעביר אותו אל המחשב האחר אך עם פרטי מקור של המחשב השולח ולא של התוקף כך שלא ידע שהמידע עבר דרך התוקף.

13. את פקטת ה-ARP שמרעילה את הסורק יש לשלוח באופן חוזר כל כמה זמן מפני שכל כמה זמן טבלת ה-ARP מתאפסת.

הרחבה לפרק קריפטוגרפיה

פרק זה משלב רקע מתחום תורת המספרים וזאת על מנת להבין כיצד מערכות הצפנה מודרניות עובדות כגון RSA.

דוגמאות לתכנים אפשריים:

- ❖ מודל שרת-לקוח המצפינים הודעות באמצעות OTP ע"י יצירת pad לכל הודעה באופן סימטרי ע"י פונקציית PRNG באמצעות הרחבת seedn (קבוע ל-2 הצדדים). ניתן להרחיב את התרגיל תוך יישום יצירת ה seed (המשמש למעשה כמפתח) לשני הצדדים "באזיר" באופן סימטרי באמצעות Markle Puzzle או בצורה א-סימטרי באמצעות RSA.
- ❖ שימוש באלגוריתם המיישם DES ולשנות את מבנה ה s-Box. באופן זה ניתן להחליש את ההצפנה. התלמיד יכול לבצע את השינוי האלגוריתמי, לנסות לנתח את חולשות המערכת ואת מרכיב הסיבוכיות של האלגוריתם החדש.
- ❖ בניית מערכת הצפנה ציבורית (א-סימטרית) תוך יישום יצירת והחלפת המפתחות דרך אלגוריתם RSA.
- ❖ ניתן להציג מערכת הצפנה אנונימית (Black-Box) המיישמות אלגוריתם E להצפנה של מחרוזות בגודל N ביטים, ואלגוריתם D לפענוח (כך שלכל מסר m ולכל צופן c קיים מפתח יחיד).
- a. על התלמיד לנסות לבנות מערכת הצפנה עבור 2 ביטים (הודעות גדולות יותר)
- b. בניית אלגוריתם המוצא את כל המפתחות האפשריים
- ❖ ניתן לשלב בין RSA ל AES, ע"י יצירת מפתח סימטרי PKI באמצעות RSA בעזרת Certificate Authority ומעבר ל AES
- ❖ פיתוח התקפת הודעה נבחרת על מערכות קפטורגפיות ידועות ולא ידועות:
 - a. לתוקף מותר לבקש פענוחים של כל קריפטוגרמה השונה מקריפטוגרמת הפיענוח הנדרשת c ולגלות את c ביעילות
 - b. התוקף מבקש פענוח של מספר מוגדר של קריפטוגרמות (שונות מ c) ולאחר מכן מסוגל ע"י אלגוריתם יעיל לפענח את c
- ❖ פיתוח אלגוריתם המיישם חתימה דיגיטלית בשיטת RSA עם מפתח ציבורי (p,A).
 - a. כעת ניתן לייצר אלגוריתם מוחלש לחתימה כך שיהיה ניתן לייצר בצורה יעילה התקפה המייצרת חתימה חוקית על מסמך אחר, ולהסביר מדוע החתימה חוקית.
 - b. כעת ניתן לבקש מהתלמיד לשנות את אלגוריתם החתימה מהסעיף הקודם כך ההתקפה מהסעיף הקודם לא תעבוד.

דוגמאות נוספות לדרכי הוראה:

- ❖ ניתן להציג אלגוריתם שיחלוף מורכב רב אלפביתי כדוגמת vigenere ופיענוח אורך המפתח באמצעות BF על התדירויות. זהו צופן מורכב לכן ניתן להשתמש בתוכנה יעודית לפריצת הצופן (לא לממש אותה), אלא יותר לתת לתלמיד להבין את מנגנון ההצפנה, הפיענוח וההתקפה.
- ❖ נכנס למערכות חתימות דיגטליות המבוססות על RSA או/וגם Lamport one-time signature ונסביר איך לבצע אוטנטיקציה על מסמכים, מהי חתימה חוקית ואיך ניתן לזייף חתימה. מומלץ לתת לתלמידים נוסחת אלגוריתם RSA ולחשוף נתון נוסף (שאמור להישאר סודי, למשל חלק ממפתח) על מנת לזייף מסמך כחוקי.
- ❖ ניתן גם לתקוף את בעיית הכפילות של RSA.
- ❖ החלשת המערכת ע"י שינוי המנגנון הפנימי שלה. לדוגמה: הורדת s-box.
- ❖ לתת לתלמידים לחשוב על חורי אבטחה במערכות מוחלשות (למשל האם תוקף יכול לבצע התקפת הודעה נבחרת והתקפת know-plaintext) ולתקן אותם.
- ❖ כניסה לתחום תורת המספרים. ללמד מבוא ופעולות בחשבון מודולרי, להסביר מהיא פונקציית gcd (ניתן להציג את האלגוריתם של אוקלידס לחישוב gcd) וכיצד ניתן לייצר מפתחות פרטיים. להסביר את המוטיבציה ביצירת פונקציות קלות לחישוב אך קשות (מאוד) לפירוק לגורמים (דרך מספרים ראשוניים). אפשר להסביר מהו שדה, קבוצת המספרים בשדה והקבוצה הזרה לשדה.
- ❖ להציג את מערכת RSA. לחשוב על דרכים להחליש אותה ותקוף אותה.

הרחבה לפרק אבטחת מערכות הפעלה

דוגמאות לתכנים אפשריים – תרגול תיאורטי:

מכונות וירטואליות מתארחות ועקיפת מנגנון encapsulation . בסביבת ענן, מכונות וירטואליות של לקוחות שונים עשויות לחלוק את אותה החומרה כלומר, לרוץ על אותו המחשב הפיסי. לעיתים קרובות, מכונות וירטואליות כאילו חולקות את שכבת המטמון L2 במעבד (מטמון זה שומר נתונים לפי כתובות פיסיות ולא וירטואליות), אשר תפקידה לחסוך במספר הגישות לזיכרון הראשי, שכן גישה כזו יקרה בערך פי 10 מגישה ל-L2. אין מגבלה על כמות הנתונים שכל מכונה וירטואלית יכולה לשמור במטמון (מלבד גודל המטמון עצמו). פתחו מנגנון המממש ערוץ חשאי בין מכונות וירטואליות החולקות את שכבת המטמון L2. כלומר, הסבירו כיצד ניתן לשדר ביט ממכונה וירטואלית אחת לשנייה בהסתמכות על המטמון בלבד.

רשימת נושאים מפורטת ללימוד:

- ❖ הסבר כללי על מבנה מערכת ההפעלה
- ❖ מערכת הפעלה: ניתן להתמקד ב windows או לינוקס לפי בחירתו של המורה.
- ❖ יש להתמקד במנגנונים המרכזיים: מערכת ניהול קבצים, ניהול הרשאות, ניהול הגדרות.
- ❖ יש להתמקד במבנה ומיפוי הזיכרון (פיסי ווירטואלי). ניתן להציג את מנגנון FAT32 ו NTFS.
- ❖ לאחר סקירת ארכיטקטורה ומרכיביה הראשיים של מערכת ההפעלה ניתן להסביר על מנגנוני האבטחה של מערכת ההפעלה.
- ❖ מומלץ להסביר את תפקידיו המרכזיים של ה kernel ודרך פעולתו.
- ❖ יש לסקור מנגנוני בקורות כניסה, sys calls , traps על מנת להגביל שימוש במשאבי מערכת קריטיים כדוגמת I/O.
- ❖ בשלב הזה ניתן לבצע התקפת BOF על מערכת protostar. מומלץ ליישם את ההתקפה דרך אתגרים וברמות קושי משתנות (כל תלמיד יכול להגיע לרמת קושי שונה).
- ❖ יש לסקור מנגנון אוטנטיקציה והרשאות (מומלץ בלינוקס שכן הוא קל יותר לצורך יישום תרגיל פרקטי בנושא זה). ההמלצה ליישם מנגנון PAM על מספר אילוצים כמו: סיסמאות, זמני כניסה ו SSH.
- ❖ יש לסקור את מנגנון System call interposition הפוגע בזדון ביישמים הרצים ב User Mode.

❖ יש לסקור את מנגנוני בידוד התהליכים במערכת ההפעלה וחשיבותם.

אופציונלי

1. נסביר את מנגנון GSWTK להגנה מפני System call interposition
2. ניכנס מעט לארכיטקטורת מערכות וירטואליות ונסביר כיצד התהליכים מבודדים בהן. כמו כן מומלץ לסקור את מנגנון ה hypervisor וכיצד מתבצעות הרשאות גישה ל kernel ב VM
3. לאחר הסבר בידוד התהליכים ניתן לסקור מתקפות כגון Meltdown, covert channels על מנת להפר את בידוד התהליכים ולנסות להעביר תקשורת אסורה בניהם

הרחבה לפרק הגנת סייבר מפני קוד עוין

הרחבה מטרות ביצועיות:

- ❖ התלמיד יסביר כיצד לאבחן Face-AV /Ransom Trick, ProcessHacker
- ❖ התלמיד יסביר כיצד לאבחן התחזות ברשת (faking network) באמצעות NCAT ו WIRESHARK
- ❖ התלמיד יסביר כיצד לנתח documents and web-pages
- ❖ התלמיד ינתח כיצד קוד לא תקין עושה שימוש ב- Windows executable image format
- ❖ התלמיד ינתח כיצד קוד לא תקין עושה שימוש לרעה בטכניקת load/mapping memory של PE
- ❖ התלמיד יסביר במילים שלו כיצד נטענים DLL עבור PE Image וכיצד DLL ממופים ע"מ להשתמש עם מודולים

דוגמאות לתכנים אפשריים:

- ❖ כתוב תכנית שמנסה למחוק את התיקיה "C: \ Windows". ודא שהתוכנית, כשהיא מופעלת על ומכונה וירטואלית. קמפל לקובץ "EXE".
 - העלה את קובץ ההפעלה ל- VirusTotal בכתובת <https://www.virustotal.com> וצפה בתוצאות. האם הקובץ תואם חתימות אנטי-וירוס קיימות?
 - הפוך שתי גרסאות נוספות להפעלה שלך, שלכל אחת מהן יש מזהה Hash שונה, העלה אותם גם ל- VirusTotal, . נסה להסביר את תוצאותיהם ולדון בהן.
- ❖ עבור קובץ התרגיל המוצפן (שימו לב, הקובץ יכול להכיל קוד זדוני ולכן יש הריצו על מכונה וירטואלית בלבד!!), נתח באמצעות הכלים שנלמדו וענה על הסעיפים הבאים:
 - לאחר פיענוח דחיסת הקובץ, האם האנטי-וירוס זיהה אותו כתוכנה זדונית? אם כן, איזה סוג של תוכנה זדונית היא? איזו פעולה נקטה Antimalware?
 - מה ערך ה- hash של הקובץ שלא מוצפן?
 - האם הקובץ תואם חתימות אנטי-וירוס קיימות?
- ❖ לגבי כל אחד מהדפדפנים הבאים: Internet Explorer, Chrome, Firefox. השתמש בכלי הניתוח הדינמי שלמדת, נסה למצוא היכן שם המשתמש וסיסמאות נשמרות, בעזרת המנגנון auto-complete mechanism נסה גם להבין באיזה פורמט הנתונים נשמרים.

❖ נתונים מספר קבצים (חלקם זדוניים וחלקם לא), נתח כל קובץ באמצעות ניתוח דינמי בסיסי. אתה רשאי להשתמש בכלי ניתוח סטטיים בסיסיים. בדוח הניתוח שלך התייחס לנקודות הבאות (אם רלוונטיות):

- האם הקובץ משמש כתוכנה זדונית?
- מה התוכנה הזדונית מנסה לעשות?
- האם היא מנסה לתפעל את מערכת הקבצים, registry או משאבים אחרים של Windows?
- האם היא מנסה ליצור child processes או לייצר תהליכי ריצה אחרים?
- האם היא יוצרת מספר מוגזם של threads?
- האם היא מנסה להריץ תוכניות קיימות במחשב היעד?
- האם היא משתמשת בפונקציות קריפטוגרפיות?
- האם היא מנסה להתקין את עצמה במחשב היעד כדי להשיג שליטה על תהליכים?
- האם היא מנסה להתחבר לאינטרנט?
- האם היא מנסה להתחבר hosts או לשרתים אחרים ברשת המקומית?

רשימת נושאים מפורטת ללימוד:

- ❖ יש להתחיל עם הסבר כללי על קוד זדוני ואילו וירוסים נפוצים קיימים.
- ❖ הכנת מעבדה - יש להתקין על כל המכונות של התלמידים virtual box ולהתקין iso של windows 7 כמכונה וירטואלית (ראה נספח)
- ❖ ניתן להציג חלקים מקודים של תוכנות זדוניות או דרכי פעולה שלהן, כגון: Rootkit, APT, Bootkit, Bots & botnets, Scareware, Spyware, Launcher, Information-stealing malware
- ❖ לאחר מכן, נציג כלי תוכנה לניתוח סטטי כמו: VirusTotal, strings, PE viewer
- ❖ כעת נכנס לכל נושא ה packing. נלמד מדוע קודים שניים במחלוקת משתמשים בזה (הקטנת הגודל, יצירת קובץ יחיד, הופך את ה strings וההוראות לבלתי קריאות)
- ❖ נכנס ל packing level (3 רמות): Archivers, Packers, Encryption/Protection tools
- ❖ נמשיך לתרגיל דוגמא לניתוח סטטי עבור קובץ זדוני וננתח אותו באמצעות כלי התוכנה שנלמדו תוך דגש על מספר מאפיינים:
- ❖ ניתוח זמן קומפילציה

- ❖ באמצעות PEid נבדוק האם עבר packing
- ❖ שימוש ב "strings" – ע"מ לנתח string חשודים. לדוגמא: אם נזהה שימוש ב-kerne132.dll, יכול להעיד כי נעשה שימוש ב dll זה כדי להזרים קוד כלשהו או להעתיק קובץ אחר אליו לשימוש נוסף (ניתן להסיק מהשימוש בפונקציות העתקה שלפניו). דוגמא נוספת ל string היא - השימוש במחרוזת "C:*" המציינת שהקובץ מנסה לפתוח (גישה לקריאה / כתיבה) של כל הקבצים בכונן c.
- ❖ כעת נכנס לניתוח דינמי ונלמד כיצד ניתן לנתח את הקוד תוך ניטור תוצאות אונליין. ניתוח מסוג זה דורש סביבה בטוחה.
- ❖ נסביר את יתרונות הניתוח הדינמי בהשוואה לניתוח הסטטי. בדינמי ניתן ממש לבחון את דרך הפעולה של הקוד הזדוני. נסביר טכניקות לצירת סביבה בטוחה: VM NAT מול BRIDGE
- ❖ כעת נכנס לכלי ניתוח דינמיים כמו: procmon, Process Explorer, ApateDNS, ZenMap,
- ❖ נמשיך עם תרגיל לדוגמא ונדגים ניתוח דינמי המתמקד ב:
 - זיהוי אופרציות בזמן ריצה
 - שינויים במערכת קבצים
 - יצירת child process
 - התחזות
 - התנהגות אנומלית של threads
 - מניפולציה על תהליכים ע"י השוואת Image to Memory strings ובדיקה עד כמה הם משתנים
- ❖ נסיים בלימוד טכניקות להזרקת DLL ובהבנת מנגנון hooking ע"י שימוש במספר API חשובים: CreateRemoteThread, WriteProceessMemory, VirtualAlloc, OpenProcess
- אופציונלי:
- ❖ נסביר מהו PE header וכיצד ניתן לעשות עליו מניפולציות (למשל איך להמיר DLL ל EXE)
- ❖ נבדוק import dll באמצעות Dependency Walker (כך ניתן להראות בדיקה של core functionality כמו גישה ומניפולציה לזיכרון, קבצים וחומרה). הבהרה: ישנם סוגי dll שנוכל לדעת מראש האם שימוש בהם בקוד עלול להפוך אותו לקוד זדוני. לדוגמא : WS_32.dll - מכיל פונקציות שקשורות לרשת, תוכנית שמייבאת את dll זה, ככל

הנראה תתחבר לרשת או תבצע משימות הקשורות לרשת

❖ ניתן להציג דוגמא, למשל ניתן לקרוא לתוכנית rundll32.exe לביצוע בינארי שרירותי

ולהדגים כיצד יריבים עשויים לנצל פונקציונליות זו על מנת לעשות Execution של

פרוקסי כדי להימנע מפעולת כלי אבטחה סטטיים ולעקוץ רשימת whitelisting

❖ ניתוח דינאמי מתקדם:

- בדיקת שליחת פקטות לא רצויות דרך NCAT ו WIRESHARK (רק לאחר

בדיקה סטטית של שימוש ב dll זדוני כדוגמת WININET.dll)

- בדיקת התנגות התהליך ב process monitor : נאבחן קבצי רישום registry

לקריאת נתוני אינטרנט נפתחים וסגורים וגם שימוש ב DLL ע"מ צור קובץ,

סגור קובץ.