

מטלה תכנותית - קורס 20940

(مم"ז 16, מבוא לאבטחת המרחב המקומי)

ניתוח השוואתי של מנגנוני אימות מובוסי סיסמאות

מבוא

בעבודה זו נתמקד בניתוח מנגנוני אימות מובוסי סיסמאות. כידוע, החזק והיעizable של מנגנוני האימות משפיעים באופן ישיר על עמידותן של מערכות בפני שיטות תקיפה נפוצות כגון Brute-Force-Spraying ו-Password-Spraying. מטרת הפרויקט היא לבדוק כיצד שיטות שונות לאחסון סיסמאות אימות משתמשים מתקדמת מול תקיפות אלו, ולנתה את היתרונות הנוספים של אמצעי הגנה שונים כגון הגבלת קצב, נעילת חשבון, CAPTCHA ו-TOTP (Time-based One-Time Password).

פרויקט זה משלב אלמנטים תיאורתיים ומעשיים. תצריכו לחקור מנגנוני הגנה ושיטות תקיפה שחלקים טרם הכרתם, ואח"כ לתוכנן, למשול ולבנות ניסוי מבוקר המדמה תקיפות אימות בסביבה מבודדת. לשם כך תאוסףו ותנתחו מבדים שונים כגון זמן-לפריצה, מספר ניסיונות, ניסיונות לשנייה ושיעור הצלחה עבור רמות חזק סיסמה שונות ותכורות הגנה מגוונות.

שימוש לב: הפרויקט מתבצע בזוגות (בלבד). כל חריגה דורשת אישור של המנהה.
חשיבות: אם קיבלתם אישור בלבד, לא יתאפשרו בקשות להקלה בסוגנון: עשית את הפרויקט בלבד וכן לא הספקתי Z Y X וכדומה.
תאריך הגשה יפורסם באתר בנפרד.

מטרות הפרויקט

מטרה מרכזית

לבצע ניסוי הניתן לשחזור, המשווה בין מנגנוני גיבוב ואימות סיסמאות (למשל: Argon2, bcrypt, SHA-256+salt), בוחן את השפעת מנגנוני ההגנה השונים כגון pepper, salt, הגבלת קצב, נעילת חשבון, CAPTCHA ו-TOTP, ובוצעו ניתוח סטטיסטי של הנתונים המתקבלים.

מטרות נוספות

- למדוד זמן-לפריצה ושיעור הצלחה תחת מגוון מנגנוני הגנה.
- לכמת את ההשפעה של כל מנגן הגנה בנפרד ובסילוב.
- להעיר את הפרשה בין השימושות והביטחונות בכל גישה.
- להפיק דוח מחקר תמציתי המסכם את הממצאים הניסויים.

תיאור המשימות

עליכם לבצע (כאמור בזוגות) את המשימות הבאות:

- תכנון והקמה של שירות אימות וירטואלי התומך במספר מגנוני אחסון סיסמאות.
- ייצור סט נתונים משתמשים בעלי סיסמאות חלשנות, בינויות וחזקota.
- סימולציה של התקיפות Brute-Force-Password-Spraying ו-Brute-Force-TOTP.
- הפעלת מגנוני הגנה נוספים (Rate-Limiting, CAPTCHA, TOTP, Pepper) וחזרה על מדידות.
- רישום וניתוח של מדיי ביצוע ותוצאות התקיפה.
- המרת כל תוצרי החoba והגשתם.

תוצרי הפרויקט

כל צוות יגיש:

- דוח מחקר (8-6 עמודים): כולל מבוא, מתודולוגיה, תוצאות, ניתוח, דיוון, שיקולים אתיים ומקורות.
- לוגים גולמיים: נתונים מובנים (CSV או JSON) של ניסיונות האימות וההתוצאות.
- קובצי קונפיגורציה: תצורה אוניברסלית של סביבת העבודה (ללא מפתחות סודיים).
- מצגת:VICOM קצר של מטרות, תצורה וממצאים.
- סרטון הדגמה (4-2 דקות): הצגת הסביבה והפתרונות המרכזיים.

הנחיות אתיות ובתיותיות

- כל הניסויים יתבצעו אך ורק על מערכות מקומיות או וירטואליות השויות לסטודנטים.
- אסור לחלוטין לבצע התקיפות על רשותות או מערכות ציבוריות.
- יש להשתמש אך ורק בנתוני משתמשים מלאכותיים ולא בנתוני אמיתי.
- הדוח חייב לכלול הצהרת עמידה בכללי האתיקה.

מושגים מרכזיים והסבירים

TOTP – Time-based One-Time Password

הגדרה: מגנון אימות דו-שלבי היוצר קודים זמינים הנזרים מסוד משותף. כל קוד תקין לפרק זמן קצר (לרוב 30 שניות).

ללא ניטות: מוסיף גורם דינמי עצמאי שמנע כמעט לחלוטין הצלחת התקיפות Brute-Force.

היבטי מימוש: שלב רישום (החלפת סוד), סנכרון זמינים (סיטית שעון), טווח סבירות בבדיקה (עד כמה השרת מוקן לקבל קוד שמקורו בזמן מעט שונה).

מזהות: RFC 6238, ומספרות נפוצות כגון 4to6d.

Password-Spraying

הגדרה: ניסיון להשתמש במספר קטן של סיסמות נפוצות על פני משתמשים רבים בńיגוד ל-Brute-Force.

הממוקד בחשבון ייחיד.

லְוּוֹנְטִית: מדגדים כיצד Rate-Limiting בرمת משתמש אינו מספיק ללא מגבלות גלובליות.

מָקוֹדִי נִתְחַזֵּק: שיעור הצלחה לחשבון, מספר ניסיונות כולל, והשפעת מדיניות ההגנה.

מקורות: Microsoft Security Research, OWASP, ובלוג OWASP.

Pepper

הגדרה: ערך סודי גלובלי הנוסף לסיסמה לפני הגיבוב ונשמר בנפרד ממseed הנתונים.

הבדל מ-Salt: הוא ייחודי למשתמש ונשמר לצד הגיבוב; Pepper גלובלי ונשמר כסוד.

ლְוּוֹנְטִית: מחזק את אחסון הסיסמות במקרה של דליפת מסד נתונים.

מקורות: OWASP Password Storage Cheat Sheet, NIST SP 800-63B.

פרוטוקול ניסוי (תהליכי מוצע)

שימוש לב: הפרוטוקול שלhalten הינו הצעה שוממלץ מאוד להיצמד אליה. אם מסיבה כלשהי אתם מעדיפים לבצע את הניסוי תוך שימוש ב프וטוקול שונה, עלייכם להגדיר אותו במדויק ולקבל אישור ממנהל הקבוצה שלכם לפני המימוש.

- הגדרת GROUP_SEED ייחודי ושילובו בקבצי התצורה.
- GROUP_SEED יהיה XOR של שני מספרי הת.ז של חברי הקבוצה.
- יצירת סט משתמשים מבוקר (סיסמות חלשות/בנייה/חזקות).
- עלייכם להגדיר מה כלל בכל קטגוריה.
- הרצת בדיקות בסיס (לא מנגנון הגנה).
- הוספת מנגןן הגנה ייחיד בכל פעם והרצה התקיפות מחדש.
- רישום תוצאות עקבי בכל התצורות.
- סיכום ממצאים בטבלאות ותרשימים.

מטרת GROUP_SEED ושימושו

מטרת עיקרייה:

- **יחידיות ושלמות:** מבטיח הבדל ווחזור בין קבוצות הסטודנטים.
- **בדיקה מקורית:** מאפשר לוודא שהניסיונו בוצע עצמאית.
- **עקבות:** מופיע בלוגים, בكونפיגורציה ובבדיקות לקישור התוצרים לקבוצה מסוימת.
- **וחזרה:** מסייע להריץ מחדש ניסויים במדויק.

הערה: GROUP_SEED אינם סוד (לא Pepper), אלא מזהה.

שימוש מומלץ:

- הכללת ה-Seed במטה-נתונים של קבצי תצורה/נתונים.
- ציון h-Seed ב-README.
- שימוש בו לשימון תרשימים או קובצי תוצאות.
- אחת מהסיסמות תהיה GROUP_SEED.

מדדים ולוגים

- מדדים נדרשים: סך ניסיונות, זמן כולל, ניסיונות לשנייה, זמן-לפריצה, שיעור הצלחה, השהיה, שימוש ב-CPU/זיכרון.
- פורמט לוגים: CSV או JSON הכול Timestamp, שם משתמש, סוג גיבוב, תוצאה (הצלחה/כשלון), מצב מנגןני ההגנה.
- תיעוד: יש לכלול תיאור קצר של אופן איסוף הלוגים וaimotם.

ניתוח נתונים וסטטיסטיקה

- **זמן תקיפה:**
 - מוצע
 - חיצון
 - אחוזונים (לדוגמה, האחוזון ה-90 הוא הזמן שמתחתיו נמצא 90% מהתקיפות)
- השוואת התפלגיות בין מנגןני ההגנה:
 - כלומר להשוות את צורת ההתפלגות של זמן התקיפה (או מדדים אחרים) בין תצורות ההגנה שונות - למשל, ללא הגנות, עם TOTP, Rate-Limiting, עם Pepper וכו'.
- **דיאן בתקוף הניסוי:**
 - מגבלות: מה היי המגבליות של הניסוי? לדוגמה: עצמת חומרה מוגבלת, מספר קטן של משתמשים, שימוש בכלים תקיפה פשוטים, זמן ניסוי קצר, הנחות פשטיות שלא מייצגות סביבה אמיתית, וכו'.
 - מקורות טעות: מה יכול היה לפגוע בבדיקה של התוצאות? לדוגמה: חוסר יציבות בראשת או בשרת, איסנכרון זמן (במיוחד ב-TOTP), טעויות לוגיקה בקוד התקיפה או ההגנה, נתונים מזוהמים או לוגים לא עקובים, וכו'.
 - שחזור הניסוי: כיצד מישחו אחר יכול להריץ את אותו ניסוי ולקבל תוצאות דומות? זה כולל: תיעוד מלא, שימוש בהונפיגורציות ברורות וחזרות, שמירה על GROUP_SEED, רשימת גרסאות של כל כל, קוד עקוב ולוגים מסודרים.

מקורות העבודה

- ניהול ההיסטוריה גרסאות (רצוי אך לא חובה ב-Git) עם התקדמות רציפה.
- צירוף ציומי מסר או קטעי וידאו קצרים של שלבי העבודה.
- תיעוד ניסיונות כושלים או לא-מכריעים.

קריטריוני הערכה

משקל	תיאור	קריטריון
25%	בahirahethtzora hanisiotit, shelita b'mashanim (kolomer shemira ul kol permatrimim kabuim be-zman, bedikat gorim achd), tivid	מתודולוגית ותכונן בהירות התצורה הניסיית, שליטה במשתנים (כלומר שמירה על כל הפרמטרים קבועים בזמן, בדיקת גורם אחד), תיעוד
20%	necnonot, ycolat shezhor, shlemot hanetonim (kolomer habtachat diok shelmot hanetonim la-ovar kol hanisiot, la-shinuyim au avodan midau)	תהליכי המימוש נכנות, יכולת שחזור, שלמות הנתונים (כלומר הבטחת דיוק ושלמות הנתונים לאור כל הניסוי, ללא שינויים או אובדן מידע)
25%	ayicot ha-sbarim, umek stutishti, pirosh hatazot	ניתוח ודיון aiocot ha-sbarim, umek stutishti, pirosh hatazot
20%	mabna, bahirahet, hazaqa, mukorot	aicot ha-dov mabna, bahirahet, hazaqa, mukorot
10%	ayicot ha-sertun / au ha-mazgat	הדגמה aiocot ha-sertun / au ha-mazgat

מבנה דוח מוצע

• ניתוח ודיון	• תקציר ומוטיבציה
• שיקולים את'יים	• ריקע
• מסקנות ועבודה עתידית	• תצורה ניסיית
• מקורות	• מתודולוגיה
• נספחים (לוגים, טבלאות, תרשימים)	• תוצאות

תכנית עבודה מוצעת

שבוע	משימה	תיאור
1	סקירת ספרות, בחירת כלים	מתווה ראשוני
2-3	הקמת שרת, ייצירת מערך משתמשים	לוגים בסיסיים
4	בדיקות Brute-Force	דוח ביןיהם
5	הוספת הגנות (Rate-Limit, Pepper, TOTP)	לוגים מעודכנים
6	ניתוח נתונים וכנתיבת דוח	טיזות דוח
7	מצגת והדגמה	הגשה סופית

טיפים מעשיים (Do / Don't)

:Do

- לשמר לוגים מסודרים, מתויגים ובעל חותמת זמן.
- לוודא סכירה זמן עבור TOTP.
- לבדוק מקרי קצה (סיסמאות חזקות, שילובי הגנות).

:Don't

- לא לשטר סיסמאות אמיות או להשתמש בשרת אימות חיצוניים.
- לא להשתמש בכל AI ללא ציון מפורש.

הרחבות אפשריות

- בדיקת השפעת פרמטרים של Argon2 (זיכרון, איטרציות).
- בדיקת ביצועים תחת מגבלות משאים (למשל, הגבלת CPU).
- שימוש התקיפות נוספת (XSS, SQLi וכו').

שימוש לב: אם אתם ממשיכם את הרחבות (או חלק מהן) יש לציין זאת במפורש בדו"ח. הדבר עשוי להעלות את הציון לפוי שיקול דעתם של המנהה. ניתן להציג הרחבות מעניינות שלא מצויינות כאן. עם זאת, הרחבות אינן תחליף ויבדקו רק אם בוצעו כל המטרות העיקריות.

נוף - הגדרות שימושיות

שימוש לב: בדומה לאמור בתיאור הפרטוקול, הגדרות וההגויות המופיעות כאן הן אופציונליות ונועדו להקל על התכנון שלכם. עם זאת כאמור לעיל מומלץ להשתמש בהן כמותה שהן.

שם פיתוח האתר ניתן להשתמש בטכנולוגיות כגון Flask ו-SQLite (או דומות להן).

הבהרה: כל הפרמטרים הקRIPTוגרפיים בפרויקט זה הם ברמת מעבדה בלבד, ומכוונים להדגמה ולמדידה. הם אינם מתאימים לשימוש בסביבת פרודקشن.

1. שטח התקיפה וסוגי התקיפות
- **שטח התקיפה:** ONLINE בלבד: REST API עם נקודות קצה `/register`, `/login`, `/login_totp`. אין גישה ל-DB dumps. אין התקיפות נוספות (XSS, SQLi וכו').
 - **סוגי התקיפות:**
 - **Brute-force** (מכoon לחישון בודד דרך קרייאות REST).
 - **Password-spraying** (מספר סיסמאות נפוצות על פניהם חשבונות רבים).
 - **Distributed spraying** (סימולציה דרך threads/VMs).
 - **אופציונלי:** התקיפות אופליאן.

הערה: למורת האמור בסעיף זה, צוות המענין לפתח סביבה וירטואלית המדמה שרת (ולא סביבה ובית) רשי'

עשויות כן.

[הכוונה היא שבמוקום לבנות ממשק ווב, הצוות יכול ליצור סימולציה של שרת באמצעות תוכנה וירטואלית או סביבה מקומית שמחקה את פעולת השרת בפועל. זה מאפשר לבדוק את המערכת בסביבה מבוקרת ללא צורך בדף או ב-UI web. כמובן, יש לציין זאת במפורש בדו"ח].

2. דרישות מינימום לשרת
 - נקודות קצה: (HTTP REST register, /login, /login_totp) (או מקביל register, /login, /login_totp).
 - תמיינה במצבי אחסון סיסמאות הנינטנים להגדרה: SHA-256+salt, bcrypt (cost=12), Argon2id (rate-limiting, lockout, CAPTCHA).
 - תמיינה בויסותים: token (באמצעות token סימולציה), TOTP (לכל משתמש), pepper (מטעינה מהסביבה ולא מאוחסן בסיס הנתונים).
 - רישום כל ניסיון אימות ל-lines (JSON CSV attempts.log) עם שדות: timestamp, .group_seed, username, hash_mode, protection_flags, result, latency_ms

3. מודל תוקף ומוגבלות
 - יכולות התוקף: ל��ח HTTP אונליין בלבד. אין פיצוח אופלין (ain dumps DB). כל ההתקפות מופנו לשרת הבדיקה.
 - ניתן להוסיף בדיקות אופלין כהרחבה.
 - מוגבלות משאים/זמן (לכל תצורת ניסוי):
 - בירית מחדר: עד 50,000 ניסיונות לכל תצורה.
 - גבול: עד 1,000,000 ניסיונות או עד 2 שעות ריצה לפי המוקדם מביניהם.
 - קיום תגבורות השירות: אם השירות מחזיר captcha_required, הלוקו צריך לפעול לפי אוטומציה ה-CAPTCHA (סעיף 5).

4. פרמטרים בירית מחדר ל-Hash
 - SHA-256 + per-user salt
 - bcrypt: cost = 12
 - Argon2id: time = 1, memory = 64 MB, parallelism = 1

הערה: השתמשו בערכים אלו כבירית מחדר להשוואה אך מומלץ לנסות ערכים נוספים.

5. אוטומציה ל-TOTP ו-CAPTCHA (איך להריץ ניסויים אוטומטיים)
 - TOTP אוטומציה: users.json יספק ב-field otp_secret לחשבונות בדיקה. סקריפטים יכולים להשתמש בסוד כדי ליצור טוקנים תקפים (קוטוק או מקבילה). לסימולציה של סנכרון זמן, ניתן להוסיף לשעון המקומי סטיה מבוקרת של ±2 שניות, להריץ את מנגן הסנכרון, ולדוח על הסטיה שהוטלה, התקון שבוצע והשgiaה הסופית.
 - סימולציה CAPTCHA: השירות לא ידרוש פתרון CAPTCHA אנושי. במקום זאת: אחרי X ניסיונות כשלים יחזיר השירות {"captcha_required":true, "captcha_token":...}.
 - לטינה אוטומטית הלוקו יכול להשיג טוקן תקף דרך endpoint admin/get_captcha_token?group_seed=<GROUP_SEED>/GROUP_SEED. שימוש בטוקן ממשיך את התקיפה האוטומטית, כך ניתן לבדוק את ההשפעה בלי חסימה אנושית.

6. GROUP_SEED (מעקב ומקורות)
 - לכל קבוצה יהיה GROUP_SEED ייחודי שהוא אמרור XOR של מס' ת.ז. של מחברי הקב'. הקבוצה חייבת להטמעו(users.json,(group_seed שדה README), ולהציגו ב-.git commit messages. המטרה: אימות מקורות ו-traceability.

7. דרישות ל-dataset
 - יצרת **30** **শিলো** ב**ডাটা**: 10 חלשים, 10 בינוניים, 10 חזקים. לטעד סיבה לשיווג של כל חשבון ב-README. אין להשתמש בנתוני משתמשים אמיתיים.
8. דרישות דיווח מינימליות
 - יש להציג, עבור כל שילוב של מצב ה-**hash** והגנות מופעלות/כביות (toggles):
קובץ **log attempts** גולמי עם timestamps ו-**GROUP_SEED**.
 - טבלת סיכום תוצאות: total_attempts, attempts_per_second, time_to_first_success (אם יש),
.success_rate_by_category
 - לחשב את הזמן הממוצע (average latency_ms) עבור כל מצב hash נפרד.
 - אם פיצוח מלא אינו מעשי בנסיבות, יש לספק **extrapolation**: חישוב זמן משוער לפיצוח על בסיס measured attempts/sec keyspace; יש לפרט הנחות.
 - הצהרה אתית קצירה המאשרת צוות לכלל הבדיקות.