digital whisper
Technologic Papers

כרטיסי ברזל: שריפת קמפיין פישינג בטכניקות לגיטימיות בזמן אמת

מאת יהודה כהן ושי נחום

הקדמה

בעידן הדיגיטלי, חיינו שזורים יותר ויותר בעולם המקוון, גם כאנשים פרטיים אנו חווים את הסכנות ממתקפות סייבר ביום יום. בין האיומים הללו, שיטות פישינג שונות מתבלטות כשיטה ערמומית, פשוטה יחסית אך יעילה במיוחד המופעלת על ידי פושעי סייבר כדי לגנוב מידע רגיש.

דמיינו לעצמכם את התרחיש הבא (סיפור אמיתי): אתם מקבלים אימייל (לתיבת GMAIL לגיטימית בשירות בתשלום של Google WorkSpace) ממה שנראה כאתר לגיטימי של UPS, המודיע לכם על משלוח ממתין הדורש תשלום קטן. אתם לוחצים על הקישור, שמובילים אתכם לדף UPS אותנטי לכאורה, כאשר אתם ממשיכים להזין את הפרטים האישיים שלכם: שם, מספר טלפון, מספר תעודת זהות, ובסופו של דבר, פרטי כרטיס האשראי. האתר אפילו מבקש לאמת את חיוב הכרטיס עם מספר הטלפון שלכם באמצעות SMS, מה שמגביר את האשליה שמדובר באתר שאין לפקפק אחרי הלגיטימיות שלו. רק לאחר אימות זה אתם מגלים את האמת המבישה: נפלתם למלכודת פישינג כרטיסי אשראי מוקפדת (Credit Card Fraud).

ההשפעה של גניבת כרטיסי אשראי היא עמוקה, בידיעה שלמרות כל אמצעי הזהירות, רימו אותך. חוויה כזו לא רק פוגעת כלכלית אלא גם פוגעת בביטחון העצמי של הקורבן, ומותירה אדם פגיע[1].

במאמר זה אנו שמחים להציג איך ניתן להפוך את הקערה על פיה, לנצח את המערכה, ועם מימוש מהיר ורחב (על ידי גורם ממשלתי/מוסדי, למשל מערך הסייבר הלאומי או חברת אשראי/בנק) ניתן אף לנצח את המלחמה כולה נגד מתקפות סייבר מהסוג הללו. חשוב לציין, כלל האמצעים שהשתמשנו בהם במחקר זה הינם לגיטימיים, תחת האילוצים של החוק הישראלי/הבינלאומי האוסר תקיפה של נכסי מחשוב ותשתיות בשום אופן.

[1] חוץ מההשפעות הכלכליות הברורות שיכולות להיות לאירוע כזה, חוקרים מראים שיש השפעה פסיכולוגית על קורבנות פישינג:
https://www.psychologytoday.com/us/blog/the-fraud-crisis/202101/how-does-fraud-impact-emotional-well-being
https://www.thesafetymag.com/ca/news/opinion/psychological-trauma-and-cybercrime/252447

המטרה שלנו הייתה ברורה: להציף את תוקף הפישינג בכל כך הרבה מידע תקין אך לא שימושי כך שהנתונים הגנובים שלו יהפכו חסרי תועלת לחלוטין. כדי להשיג זאת, השתמשנו בשילוב של כלים טכניים ואסטרטגיות חכמות, והבטחנו שהפעולות שלנו יישארו אנונימיות ויעילות. זהו סיפור הניצחון הדיגיטלי שלנו, סיפור על שימוש ביכולות בסיסיות של מגני סייבר שפשוט לא מוכנים להפסיד במלחמה... (מתקפת הפישינג של התוקף התרחשה בישראל והצליחה להפיל מספר רב של קורבנות, כאשר מחקר זה בוצע בהתנדבות).

למרות המגוון הרחב של פתרונות המיועדים למניעת פישינג, בעיית הפישינג נותרת אתגר מתמשך בעולם הסייבר. מערכות אבטחת מידע מתקדמות, כגון סינון דוא"ל, אימות רב-גורמי (MFA), ומערכות לזיהוי והתרעה, אמנם מצליחות להפחית את היקף המתקפות, אך אינן מסוגלות למגר את התופעה לחלוטין. אחת מהסיבות לכך טמונה בעובדה שבני האדם הם החוליה החלשה ביותר במערכת האבטחה. טעויות אנוש הן בלתי נמנעות, וכל אדם, לעיתים, אף ללא קשר לרמת ההכשרה או היכרותו עם איומי הפישינג, עלול ליפול קורבן למתקפה בסיסית או מתוחכמת.

מתוך הבנה זו, עולה הצורך לפתח גישות חדשות ויצירתיות להתמודדות עם הפישינג. הצפת מסד הנתונים של התוקפים יכולה להוות גישה משלימה וחדשנית בהגנה על ארגונים/חברות ואנשים פרטיים. הרעיון הוא לנצל את אותם כלים וטכניקות שבהם משתמשים התוקפים, אך לעשות זאת במסגרת חוקית ולגיטימית. בצורה זו, ניתן להרתיע תוקפים ולגרום להם להבין שמתקפותיהם לא יעברו ללא תגובה, גם אם השיגו מספר הישגים ראשוניים, וכי הם יצטרכו להתמודד עם פעולות הגנה פרואקטיביות.

השימוש במתקפה חוזרת אינו נועד להחליף את הפתרונות הקיימים, אלא להוסיף שכבת הגנה נוספת. בצורה זו, ארגונים יכולים להראות לתוקפים שהם מוכנים להתמודד עם האיום ולא רק להגן עליו בצורה פסיבית וסטטית, המגן מתחיל לנוע במרחב ומשתמש באמצעים ויכולות על מנת לנצל את יתרונות ההגנה שלו. גישה זו תורמת לשיפור המודעות והיכולת של הארגון להתמודד עם איומי פישינג, ומעודדת חשיבה מתקדמת ויצירתית במאבק נגד Cyber Crimes למיניהם.

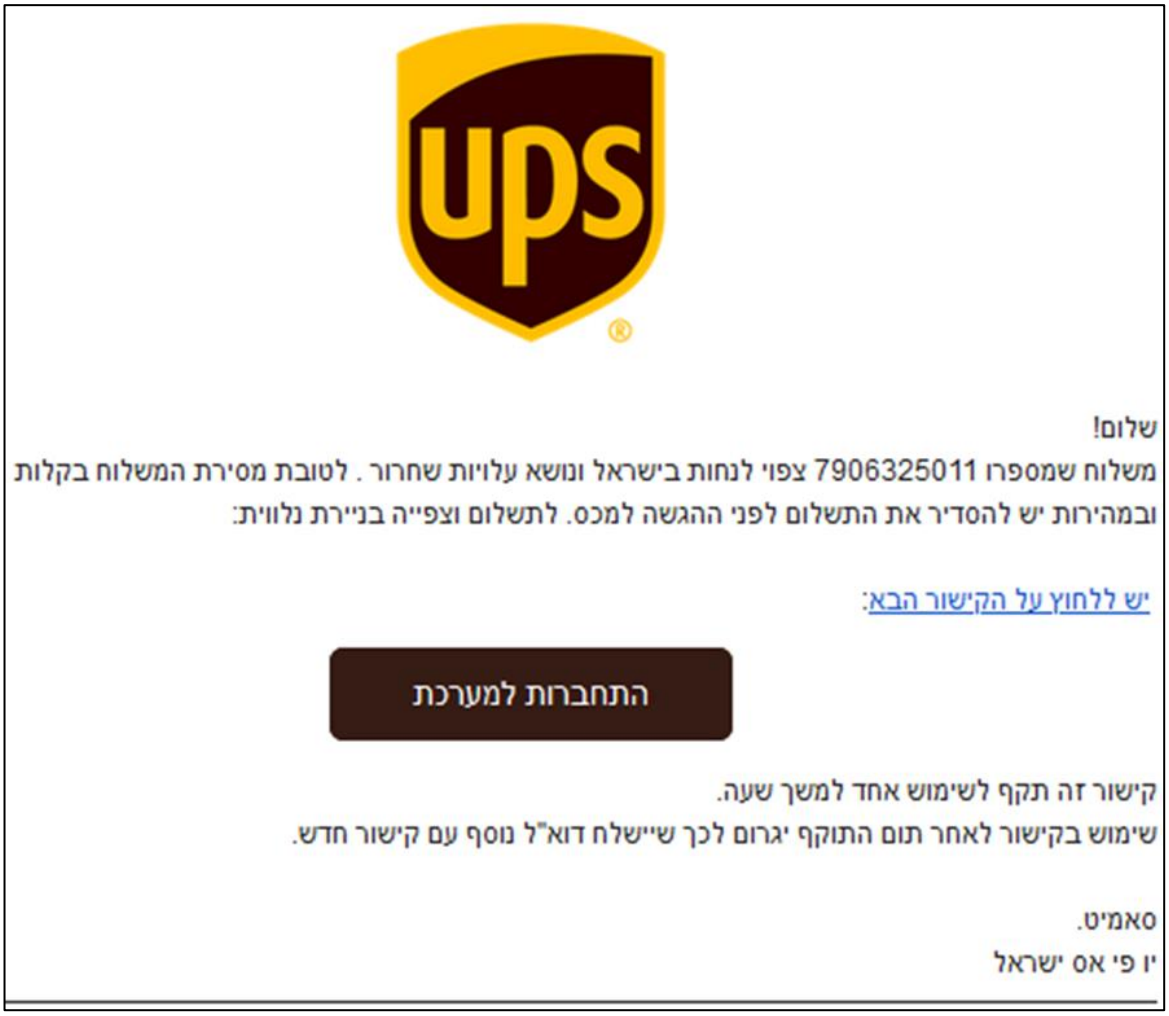
המאמר מחולק לנושאים הבאים באופן מודולרי:

- תיאור מתקפת הפישינג של התוקף
- יצירת רשימת כרטיסי אשראי וחימוש יכולות
- תיאור תהליך הסליקה ושלביו השונים
- שריפת מערך התקיפה של התוקף
- השפעה על התוקף
- סיכום ועבודה עתידית

הפישינג

פישינג הוא סוג של מתקפת סייבר שבה התוקפים מתחזים לישויות לגיטימיות כדי להערים על אנשים לחשוף מידע רגיש, כגון שמות משתמש, סיסמאות, פרטי כרטיסי אשראי ועוד. לעתים קרובות זה מתחיל באימייל או בהודעה שנראה כי מגיעים ממקור מהימן - כמו בנק, אתר סחר אלקטרוני, או, במקרה הזה, חברת לוגיסטיקה ידועה כמו UPS.

בתרחיש שאנו בודקים, מתקפת ההתחזות מעוצבת בקפידה כדי לחקות את אתר האינטרנט של UPS. זה מתחיל באימייל תמים לכאורה המודיע לנמען על משלוח ממתין הדורש תשלום קטן (המייל הגיע ל-INBOX של תיבת GMAIL WORKSPACE):



שלום!
משלוח שמספרו 7906325011 צפוי לנחות בישראל ונושא עלויות שחרור . לטובת מסירת המשלוח בקלות ובמהירות יש להסדיר את התשלום לפני ההגשה למכס. לתשלום וצפייה בניירת נלווית:

יש ללחוץ על הקישור הבא:

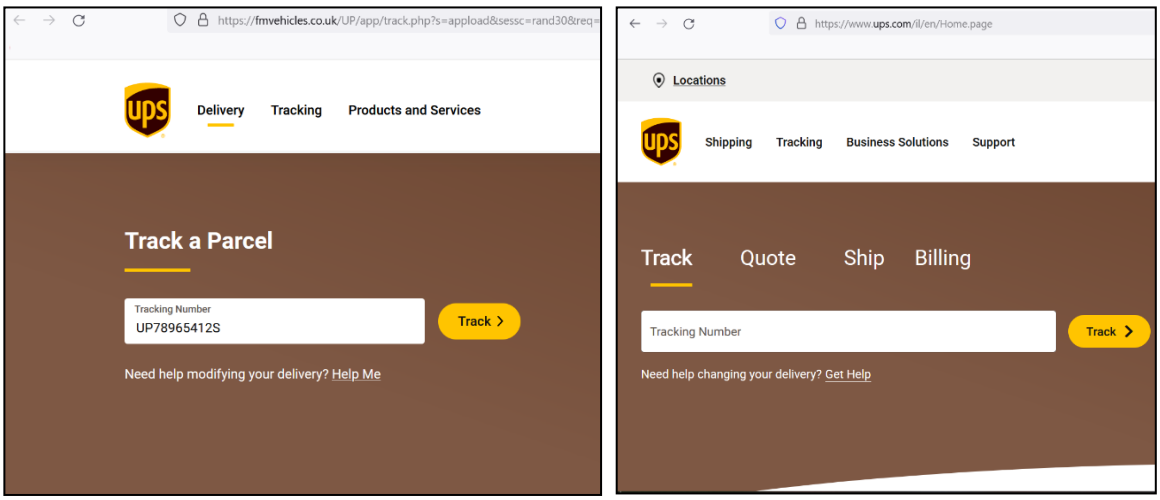
התחברות למערכת

קישור זה תקף לשימוש אחד למשך שעה.
שימוש בקישור לאחר תום התוקף יגרום לכך שיישלח דוא"ל נוסף עם קישור חדש.

סאמיט.
יו פי אס ישראל

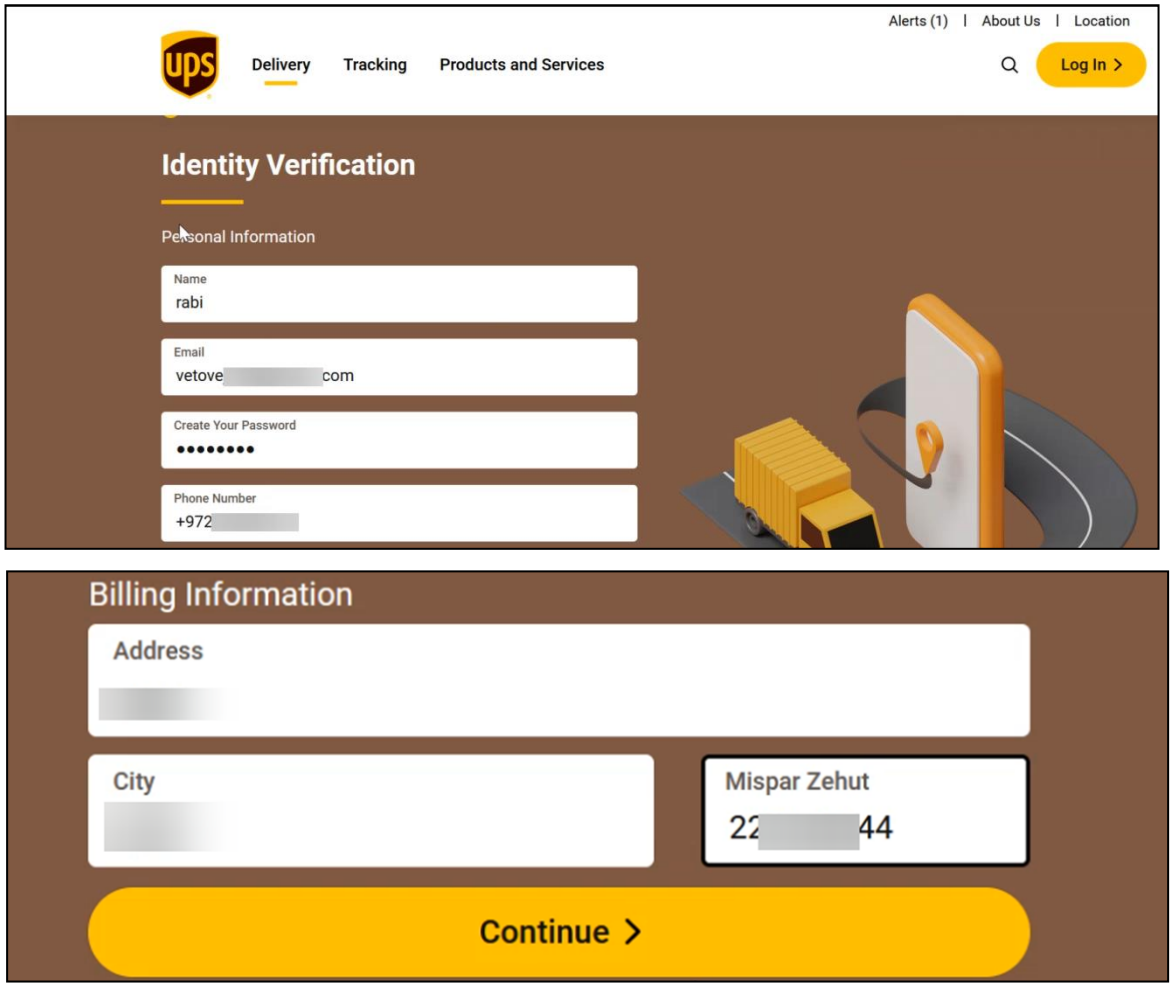
האימייל מכיל קישור שמפנה את הקורבן לדף UPS מזויף בדומיין הבא fmvehicles[.]co[.]uk, שנבנה בצורה שלא ניתן להבחין בו מהאתר הלגיטימי.

בתמונות להלן ניתן לראות את ההשקעה בפרטים קטנים של האתר כדי שידמה כמה שיותר ל-UPS:



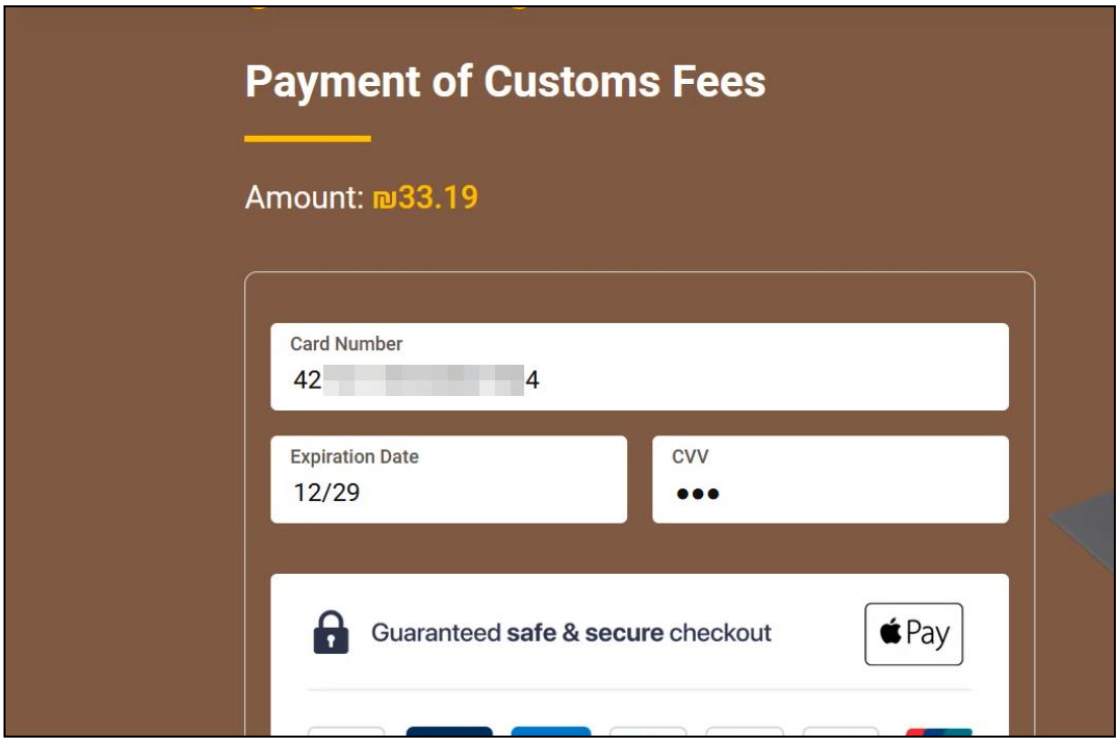
[איור 1: האתר הלגיטימי [ימין] לעומת האתר המזוייף [שמאל] - אתם משובנעים? תסתכלו שוב ותראו שעבדנו עליכם המזוייף מימין והאמיתי משמאל נא לעיין ב-URL]

האתר המזויף מבקש מהקורבן להזין פרטים אישיים, לרבות שמו, מספר הטלפון, מספר תעודת הזהות ופרטי כרטיס האשראי שלו:



[איור 2: הזנה של פרטים אישיים וכתובת]

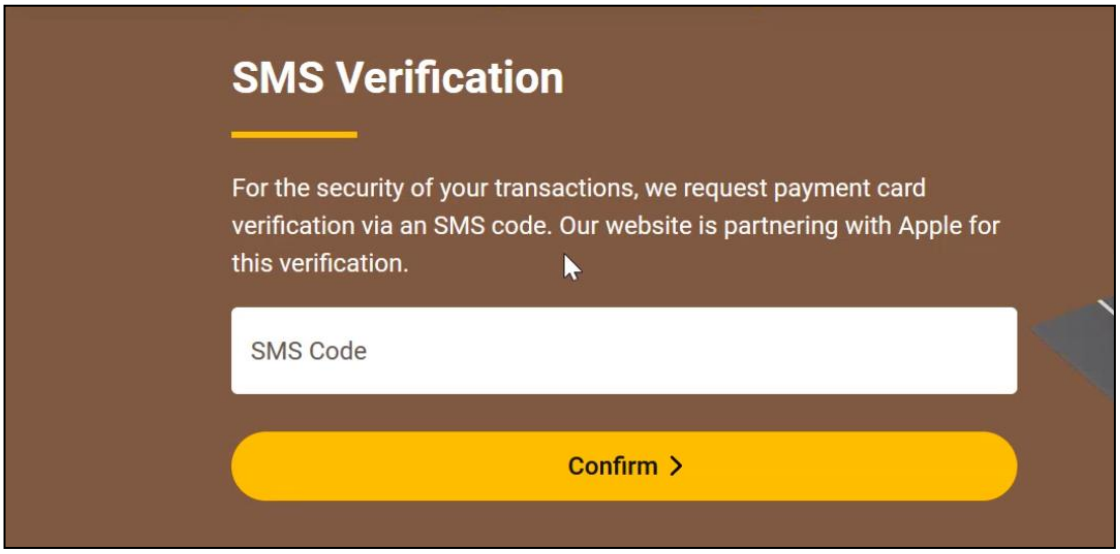
באיור הבא ניתן לצפות בבקשה להזין פרטי כרטיס אשראי



[איור 3: הזנת פרטי אשראי]

כדי להוסיף שכבת אמינות, האתר אפילו מבקש אימות SMS באמצעות הטלפון (Payment Card Verification), המחקה את האימות הרב-גורמי המשמש שירותים מקוריים רבים:

[איור 4: טעינה של הזדהות MFA]

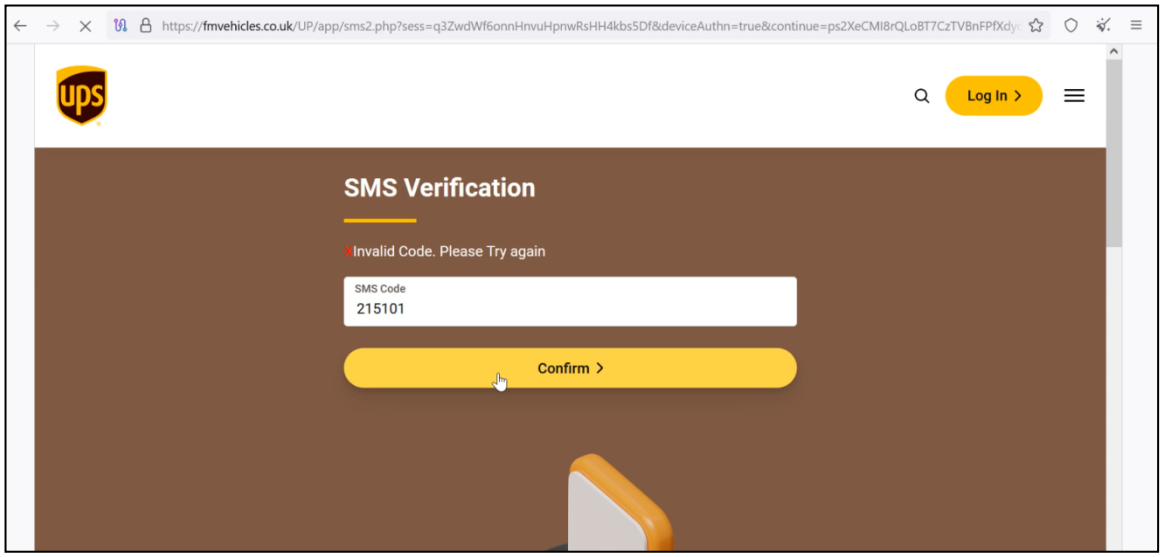


[איור 5: קבלת קוד ב-SMS]

ה-SMS לא נשלח על ידי התוקף, אלא הוא משתמש ב-API של חברות האשראי עצמן כפי שניתן לראות ב-SMS שקיבלנו במהלך הניסוי:

[איור 6: קבלת הקוד ב-SMS]

לאחר מכן הוא בודק את הקוד שהוזן לתוכו:



[איור 7: קבלת שגיאה אם הקוד אינו תקין]

הפישינג שהורץ על ידי התוקף במקרה הזה הוא פישינג איכותי מאד, מכמה סיבות:

- עיצוב מקצועי ומדויק: האתר המזויף נראה כמעט זהה לאתר האמיתי של UPS
- שימוש בטכניקות הנדסה חברתית: התוקפים מנצלים את הדחיפות והאמינות המיוחסים להתראות משלוח, כדי להפעיל לחץ על הקורבנות להזין את פרטיהם.
- אימות SMS: השימוש באימות SMS נותן לתוקף מראה של אמינות נוספת, ומטעה את הקורבנות לחשוב שמדובר בהליך אבטחה לגיטימי.
- יש כאן גניבה/ניצול כפול, גם גונבים את פרטי כרטיס האשראי וגם מחייבים אותו.

תוצאות המתקפה וניצול פסיכולוגיית הקורבן

ערכת פישינג זו מנצלת את האמון שמשתמשים נותנים במותגים מוכרים ואת הדחיפות הקשורה לרוב בהתראות משלוח. קורבנות, שנתפסו על ידי המראה המשכנע והרצון לפעול במהירות ולסיים עם המשלוח שהזמינו, נופלים קורבן להונאה. לאחר הזנת המידע, הוא נקלט מיד על ידי התוקפים, אשר לאחר מכן משתמשים בו לפעילויות הונאה או מוכרים אותו ברשת האפלה.

התקפות פישינג כמו זו הן יעילות ביותר מכיוון שהן ממנפות טכניקות של הנדסה חברתית כדי לתמרן קורבנות לביצוע פעולות שאחרת היו נמנעים מהם. התוקפים מנצלים את הפסיכולוגיה האנושית - אמון, דחיפות, פחד - מה שהופך את זה למאתגר אפילו לאנשים הזהירים ביותר לזהות את ההונאה עד שיהיה מאוחר מדי.

הבנת המכניקה של תוכניות פישינג כאלה היא חיונית לתכנון אמצעי נגד יעילים. במקרה שלנו, ההבנה של ההונאה הציתה רצון לבסס שיטה לעצור את מתקפת הפישינג ולמנוע קורבנות נוספים. הקורבן אמנם דיווח על התקרית, חסם כרטיס אשראי והמשיך הלאה, אנחנו החלטנו לבצע מחקר, תוך שימוש ביכולות טכניות כדי לשבש את פעולות התוקפים ולהפוך את הנתונים הגנובים שלהם לחסרי תועלת.

האנונימיות והפן הדיגיטלי הבינלאומי של פעולות פישינג מקשים על רשויות אכיפת החוק לטפל, לאתר ולהעמיד לדין את העבריינים במהירות. חסינות זו לכאורה רק מעצימה את התסכול של הקורבנות.

אחד מההישגים הנדרשים של המחקר הייתה למצוא שיטה לעצור את קמפיין התקיפה הזה באופן חוקי ואתי מבלי להשתמש בחולשות תוכנה, טכניקות חדירה או כל פעולה אופנסיבית שחוצה את הגבול.

הצפת כרטיסי אשראי והתחמשות ביכולות

התוכנית שגיבשנו הייתה פשוטה: להציף את התוקף במבול של פרטי כרטיס אשראי חוקיים אך לא תקינים, מה שהופך את הנתונים הגנובים שלו לחסרי תועלת. על ידי הצפת מסד הנתונים שלהם בתוצאות חיוביות שגויות, יכולנו להפוך את זה כמעט לבלתי אפשרי עבורם להבחין איזה מידע הוא אמיתי ואיזה לא. גישה זו לא רק תפריע לתוקפים - היא תחבל בכל המודל העסקי שלהם.

הצעד הראשון בתוכנית שלנו היה ליצור מספר רב של מספרי כרטיסי אשראי. לשם כך השתמשנו עם אלגוריתם luhn, השיטה שבה משתמשים לאימות מספרי כרטיסי אשראי. אלגוריתם זה יאפשר לנו ליצור מספרים לא תקינים שנראים לגיטימיים, ויבטיח שהתוקפים לא יצליחו לברור אותם מפרטי כרטיסי האשראי הגנובים.

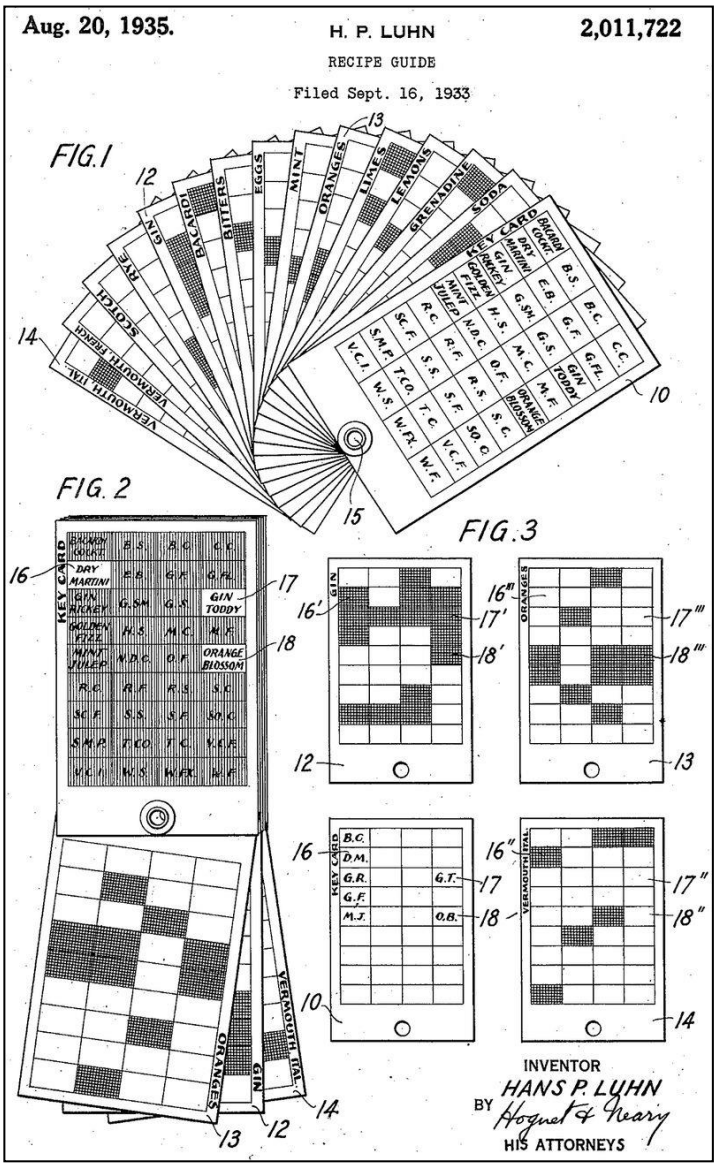
בשלב הבא, היינו צריכים דרך להעביר את המספרים האלה לתוקף באופן אוטומטי. זה הצריך שימוש בכלים שיכולים להפוך את התהליך לאוטומטיים ולהסוות את הזהות שלנו. בחרנו ב-Burp Suite, כלי רב עוצמה שמשמש בדרך כלל בודקי חדירות לבדיקות של אתרים ואפליקציות, כדי לנהל את הנתונים.

על מנת לשמור על אנונימיות ולהימנע מחסימה, חיברנו את Burp Suite לרשת Tor, וכן לבצע rotation על כתובת ה-IP שלנו ולהתחמק מזיהוי.

שימוש בסקריפט עם אלגוריתם לוהן

אלגוריתם Luhn, הידוע גם כאלגוריתם "modulus 10" או "mod 10", הוא נוסחת בדיקה פשוטה המשמשת לאימות מספרי זיהוי שונים, כגון מספרי כרטיסי אשראי. פותח על ידי מדען שעבד בחברת IBM בשם הנס פיטר לוהן בשנת 1960, לוהן היה שייך לעולם ה-"האקרים" עוד לפני שהתחילו להיות כאלו בכלל, ואפילו לפני העידן הדיגיטלי.

בשנת 1933 - בסוף תקופת היובש בארה"ב, לוהן רשם פטנט על מרשם ליצור משקאות אלכוהוליים מחומרים שהיו מותרים בתקופת היובש:



[איור 8: הפטנט של לוהן על משקאות אלכוהוליים]

בהיכרותו עם הטריקים שמשמשים אנשים לא מהוגנים, החליט לוהן למצוא אלגוריתם שנועד להגן מפני שגיאות מקריות ולא התקפות זדוניות, מה שהופך אותו לאידיאלי להפקת מספרי כרטיסי אשראי.

להלן הסבר מפורט כיצד פועל אלגוריתם Luhn[2], בואו נבין את האלגוריתם עם דוגמה:

ניקח את הדוגמה של מספר חשבון "79927398713".

- **שלב 1** - החל מהספרה הימנית ביותר, מכפילים כל ספרה שנייה:

7	9	9	2	7	3	9	8	7	1	3
	x2		x2		x2		x2		x2	
	18		4		6		16		2	

- **שלב 2** - אם הכפלה של מספר מביאה למספר דו ספרתי כלומר גדול מ-9 (לדוגמה, 6 × 2 = 12), נחבר את הספרות של תוצאת הכפל (לדוגמה, 12: 1 + 2 = 3, 15: 1 + 5 = 6), כדי לקבל מספר חד ספרתי:

7	9	9	2	7	3	9	8	7	1	3
	x2		x2		x2		x2		x2	
	18		4		6		16		2	
	9		4		6		7		2	

- **שלב 3** - כעת ניקח את סכום כל הספרות:

7	9	9	2	7	3	9	8	7	1	3
	x2		x2		x2		x2		x2	
	18		4		6		16		2	
7	9	9	4	7	6	9	7	7	2	3

- **שלב 4** - נשתמש עם פעולת ה-modulo, אם חיבור כל המספרים ב-mod10 שווה ל-0 (אם סך הכל מסתיים באפס) אז המספר תקף לפי נוסחת Luhn; אחרת זה לא תקף:

7	9	9	2	7	3	9	8	7	1	3
	x2		x2		x2		x2		x2	
	18		4		6		16		2	
7	9	9	4	7	6	9	7	7	2	3

7 + 9 + 9 + 4 + 7 + 6 + 9 + 7 + 7 + 2 + 3 = 70

מכיוון שהסכום הוא 70 שהוא כפולה של 10, ייתכן שמספר החשבון תקף.

[2] לפי https://en.wikipedia.org/wiki/Luhn_algorithm

הסקריפט שנותן כרטיסי אשראי אקראיים

בעת יצירת מספרי כרטיסי אשראי, אלגוריתם Luhn מבטיח שהמספרים יעברו בדיקות אימות ראשוניות המשמשות מערכות רבות. זה גורם להם להיראות לגיטימיים ופחות סביר שיסומנו מיד כלא חוקיים. כעת נראה את הסקריפט - הפשוט - שכתבנו וכיצד זה עובד:

```
import random
from datetime import datetime, timedelta

# Function to generate a valid credit card number using the Luhn algorithm
def luhn_algorithm(prefix, length):
    number = [int(x) for x in prefix]
    while len(number) < (length - 1):
        digit = random.randint(0, 9)
        number.append(digit)

    sum_ = 0
    for i in range(len(number)):
        digit = number[len(number) - 1 - i]
        if i % 2 == 0:
            digit = digit * 2
            if digit > 9:
                digit -= 9
        sum_ += digit

    check_digit = (10 - (sum_ % 10)) % 10
    number.append(check_digit)
    return ''.join(map(str, number))

# Function to generate a random expiration date
def generate_expiration_date():
    start_date = datetime.now()
    end_date = start_date + timedelta(days=365*5)  # 5 years into the future
    expiration_date = start_date + (end_date - start_date) * random.random()
    return expiration_date.strftime('%m/%y')

# Function to generate a random CVV
def generate_cvv():
    return f'{random.randint(100, 999)}'

# Generate credit card data
def generate_credit_card_data(count):
    numbers = []
    expirations = []
    cvvs = []
    prefixes = ["4539", "4556", "4916", "4532", "4929", "4485", "4716", "4"]  # Visa
prefixes
    for _ in range(count):
        prefix = random.choice(prefixes)
        card_number = luhn_algorithm(prefix, 16)
        expiration_date = generate_expiration_date()
        cvv = generate_cvv()
        numbers.append(card_number)
        expirations.append(expiration_date)
        cvvs.append(cvv)
    return numbers, expirations, cvvs

# Write the generated data to text files
def write_to_txt(filename, data):
    with open(filename, mode='w') as file:
        for item in data:
```

```
            file.write(f"{item}\n")

# Number of credit card entries to generate
num_entries = 100

# Generate data
numbers, expirations, cvvs = generate_credit_card_data(num_entries)

# Write data to text files
write_to_txt('credit_card_numbers.txt', numbers)
write_to_txt('expiration_dates.txt', expirations)
write_to_txt('cvvs.txt', cvvs)

print(f"{num_entries} credit card entries have been generated and saved to
'credit_card_numbers.txt', 'expiration_dates.txt', and 'cvvs.txt'.")
```

להלן הסבר על הסקריפט:

ראשית [שורות 1-2] מייבאים ספריות כמו random ליצירת מספרים אקראיים ו-datetime בשביל לייצר תאריכי תפוגה.

בשורות 4-18 יש יישום של אלגוריתם לוהן:

- שורה 4: הפונקציה luhn_algorithm מוגדרת, תוך שימוש ב-prefix (ספרות ההתחלה של הכרטיס) ו-length (האורך הכולל של מספר הכרטיס) כפרמטרים.
- שורה 5: ממירה את מחרוזת הקידומת לרשימה של מספרים שלמים.
- שורות 6-8: ממלא את הרשימה בספרות אקראיות עד שהיא מגיעה לאורך הרצוי מינוס אחת (משאיר מקום לספרת הסימון).
- שורות 9-14: מיישמת אלגוריתם Luhn כדי לחשב את ספרת הסימון:
  - מכפיל כל ספרה שנייה מימין.
  - אם תוצאת ההכפלה גדולה מ-9, הוא פוחת ממנה 9.[3]
  - מסכם את כל הספרות המעובדות.
- שורות 15-17: מחשב את ספרת הסימון (הספרה האחרונה של מספר הכרטיס) ומצרף אותה לרשימה.
- שורה 18: מצטרפת לרשימת הספרות למחרוזת אחת ומחזירה אותה.

שורות 20-26: הפקת תאריכי תפוגה:

- שורה 20: הפונקציה gener_expiration_date מוגדרת.
- שורה 21: מגדיר את תאריך ההתחלה לתאריך ולשעה הנוכחיים.
- שורה 22: מגדיר את תאריך הסיום לחמש שנים מהתאריך הנוכחי.
- שורה 23: יוצרת תאריך אקראי בין תאריכי ההתחלה והסיום.

[3] זה אותו דבר כמו לחבר את הספרות [12 באלגוריתם לוהן הופך ל-3 ניתן להגיע לזה על ידי חיבור של 2+1 או חיסור 9 מ-12]

- שורה 24: פורמט את התאריך כ-MM/YY ומחזיר אותו.
- שורות 28-30: הפקת CVVs
- שורות 32-46: הפקת נתוני כרטיסי אשראי
- שורה 32: הפונקציה gener_credit_card_data מוגדרת, לוקחת ספירה (מספר כרטיסי האשראי להפקה) כפרמטר.
- שורות 33-35: מאתחל רשימות ריקות כדי לאחסן מספרי כרטיסים שנוצרו, תאריכי תפוגה ו- CVVs.
- שורה 36: מגדירה רשימה של קידומות ויזה נפוצות.
- שורות 37-44: עבור כל כרטיס שייווצר:
  - שורה 38: בוחר קידומת אקראית מהרשימה.
  - שורה 39: יוצרת מספר כרטיס חוקי באמצעות אלגוריתם Luhn.
  - שורה 40: יוצרת תאריך תפוגה אקראי.
  - שורה 41: יוצרת CVV אקראי.
  - שורות 42-44: מוסיף את הנתונים שנוצרו לרשימות המתאימות.
- שורה 45: מחזירה את רשימות מספרי הכרטיסים, תאריכי התפוגה וה-CVVs.

הפלט של הסקריפט הינו 3 מסמכי txt, הראשון הוא credit_card_numbers.txt, השני הינו expiration_dates.txt והשלישי cvvs.txt.

מעולה אז יש לנו קוד שמחולל לנו פרטי כרטיסי אשראי, קדימה לשלב הבא.

שימוש ב-Burp Suite ותיאור הכלי intruder

Burp Suite הוא סורק רב עוצמה של פגיעות באתרים ואפליקציות וכלי לבדיקת חדירה בשימוש נרחב על ידי אנשי אבטחת מידע. הוא מספק פונקציות שונות, כולל proxy וכן קליטת כל תקשורת ה-HTTP וה-HTTPS, סורק ופורץ. הכלי Intruder בתוך Burp Suite שימושי במיוחד עבור אוטומציה של התקפות מותאמות אישית על אפליקציות ואתרי אינטרנט, כגון כוח גס, מטושטש ומניפולציה של פרמטרים.

לכידת בקשה

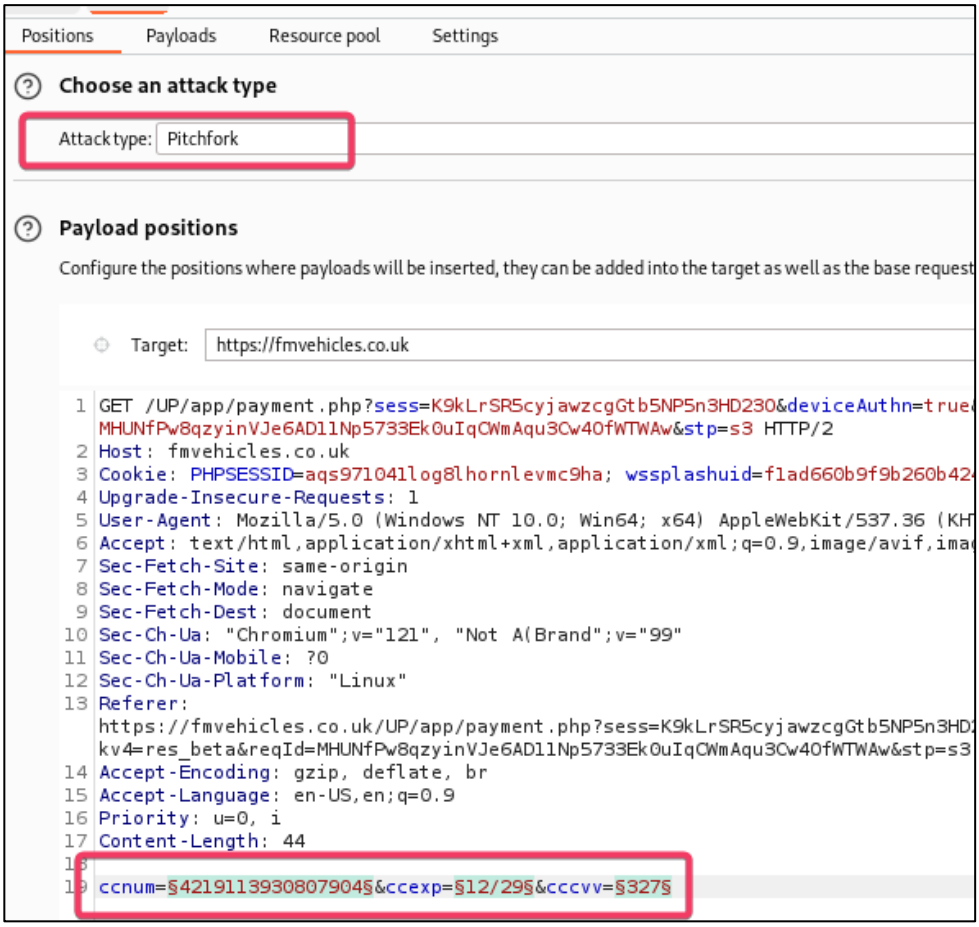
הפעלנו את ה-proxy של Burp Suite עם דפדפן האינטרנט שלו, ביקרנו באתר הפישינג ועברנו את השלבים עד לנקודה שבה יש צורך להזין את פרטי כרטיס האשראי. לכדנו את בקשת ה-HTTP המכילה את פרטי כרטיס האשראי באמצעות ה-Burp Suite Proxy:

Request

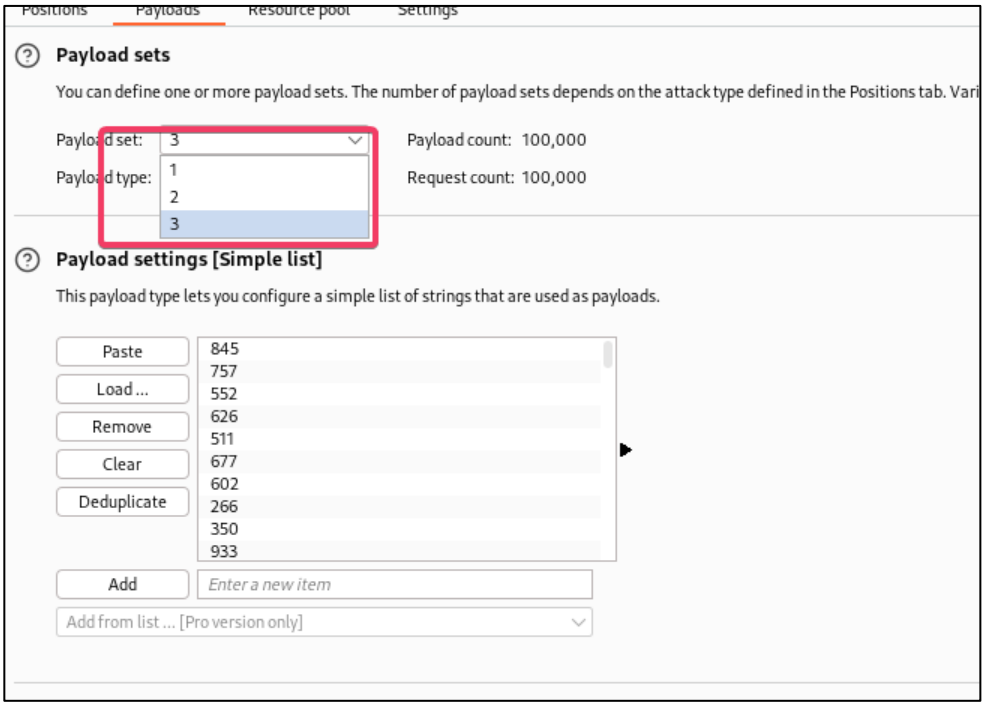
P Raw Hex

```
1  GET /UP/app/payment.php?sess=K9kLrSR5cyjawzcgGtb5NP5n3HD23O&deviceAuthn=true&continue=
   a4gr3tQezyI4HpFVcytpvVxhEBEWl4o4DwurVuXVdy5xEPGxh5LdZ9US5ffX7jEXngIlyAMdfI7z20ZxpXc4JXJ8
   i92jnhLwXkv4=res_beta&reqId=MHUNfPw8qzyinVJe6AD11Np5733Ek0uIqCWmAqu3Cw4OfWTWAw&stp=s3
   HTTP/2
2  Host: fmvehicles.co.uk
3  Cookie: PHPSESSID=aqs971041log8lhornlevmc9ha; wssplashuid=
   f1ad660b9f9b260b4244039a2dd620216c94cef4.1717683630.1
4  Upgrade-Insecure-Requests: 1
5  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
   Gecko) Chrome/121.0.6167.85 Safari/537.36
6  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*
   /*;q=0.8,application/signed-exchange;v=b3;q=0.7
7  Sec-Fetch-Site: same-origin
8  Sec-Fetch-Mode: navigate
9  Sec-Fetch-Dest: document
10 Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"
11 Sec-Ch-Ua-Mobile: ?0
12 Sec-Ch-Ua-Platform: "Linux"
13 Referer:
   https://fmvehicles.co.uk/UP/app/payment.php?sess=K9kLrSR5cyjawzcgGtb5NP5n3HD23O&deviceAu
   thn=true&continue=a4gr3tQezyI4HpFVcytpvVxhEBEWl4o4DwurVuXVdy5xEPGxh5LdZ9US5ffX7jEXngIlyA
   MdfI7z20ZxpXc4JXJ8i92jnhLwXkv4=res_beta&reqId=MHUNfPw8qzyinVJe6AD11Np5733Ek0uIqCWmAqu3Cw
   4OfWTWAw&stp=s3
14 Accept-Encoding: gzip, deflate, br
15 Accept-Language: en-US,en;q=0.9
16 Priority: u=0, i
17 Content-Length: 44
18
19 ccnum=4219113930807904&ccexp=12/29&cccvv=327
```

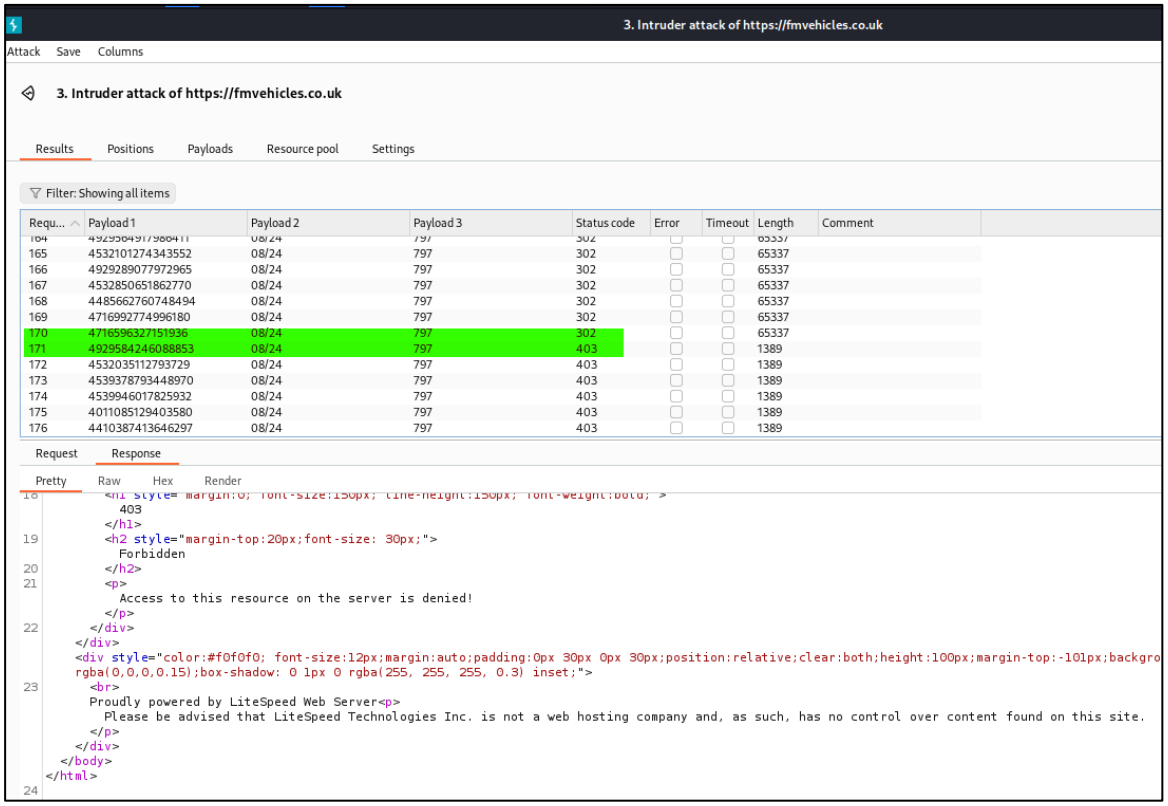
את הבקשה שלחנו אל Intruder וכדי לשלוח את שלושת הדרישות - מספר אשראי, תאריך תפוגה ו-CVV השתמשנו בסוג התקיפה pitchfork שמאפשר לשלוח כמה סוגי payloads למיקומים שונים בבקשה:



הכנסנו 3 payloads עם התוצאות של הסקריפט:



כמעט הכל מוכן להרצת התגובה על מערך התקיפה של התוקפים, אך האם אנחנו באמת מוכנים? לאחר הרצה קצרה התברר לנו שהתוקפים יותר מתוחכמים, לאחר 170 חבילות התברר שנחסמנו והתחלנו לקבל תגובה של 403 שהיא תגובה שאין לנו הרשאות יותר לשלוח חבילות לאתר וקיבלנו Access Denied:



לכן חייבים להוסיף דבר חשוב: עלינו להימנע מגילוי או חסימה על ידי התוקף.

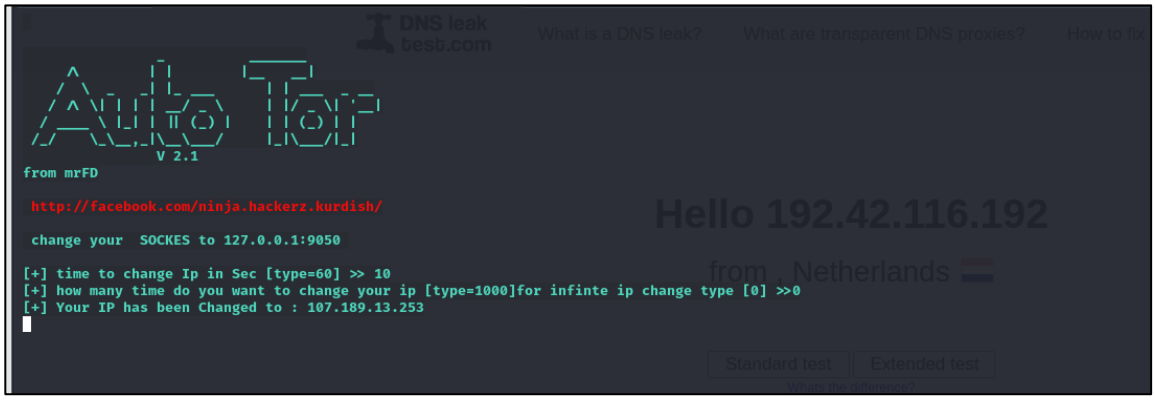
שימוש באנונימיות (TOR)

על מנת להתחמק מהאפשרות שניחסם על ידי התוקף, השתמשנו ביכולות של TOR, פלוס שינוי כתובת ה-IP שלנו. ראשית כל התקנו את Tor, ויש לנו תקשורת דרכו:

```
┌──(kali㉿kali)-[~/Auto_Tor_IP_changer]
└─$ sudo netstat -tulpn | grep tor
tcp        0      0 127.0.0.1:9050          0.0.0.0:*               LISTEN      619369/tor

┌──(kali㉿kali)-[~/Auto_Tor_IP_changer]
└─$
```

לאחר מכן התקנו את הכלי Auto_Tor_IP_Changer[4]:



לאחר הרצת הכלי ניתן להשתמש בשלל האתרים שמציגים את כתובת ה-IP ממנה הגיעה הבאה, לדוגמא: https://api.ipify.org/

וניתן לראות את כתובת ה-IP משתנה כל הזמן.

כעת יש לנו שירות של Tor בפורט 9050, אנחנו צריכים לחבר את Burp שחוץ מזה שהוא עובד עם Proxy בפורט 8080 כדי לתפוס בקשות ולשלוח תגובות ב-HTTP, שיעביר את כל התקשורת הזאת דרך השירות של Tor. בשביל זה יש לנו את SOCKS.

SOCKS (Socket Secure) הוא פרוטוקול אינטרנט המנתב מנות רשת בין לקוח לשרת דרך שרת proxy. הפרוטוקול socks פועל ברמה נמוכה יותר מ-proxy ב-HTTP, כלומר הוא יכול להתמודד עם כל סוג של תעבורה (TCP או UDP) ואינו מוגבל לתעבורת HTTP בלבד. זה הופך את SOCKS למגוון יותר עבור יישומים שונים.

Tor מספקת ממשק פרוקסי SOCKS, המאפשר ליישומים התומכים ב-SOCKS לשלוח את התעבורה שלהם דרך רשת Tor. על ידי הגדרת Burp Suite לשימוש ב-proxy SOCKS זה, כל הבקשות מ-Burp Suite עוברות אנונימיות באמצעות Tor.

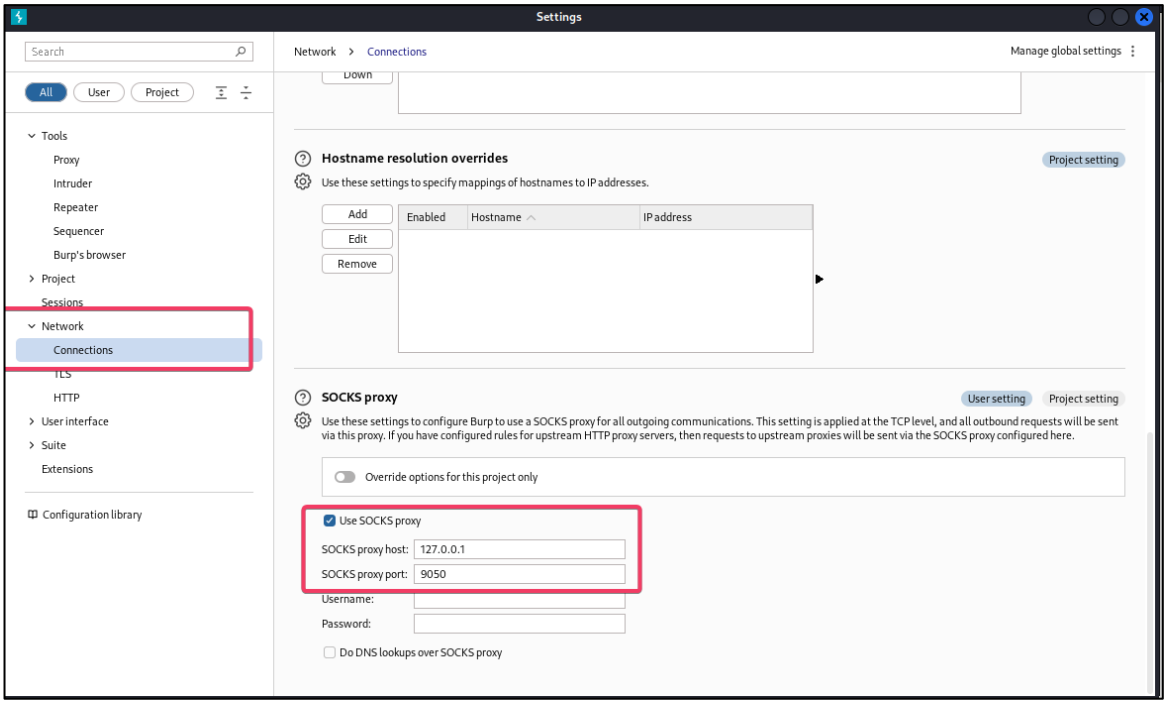
[4] https://github.com/FDX100/Auto_Tor_IP_changer - שימו לב שאתם משתמשי root כאשר אתם מתקינים את הכלי.

להלן היתרונות בשימוש ב-Tor:

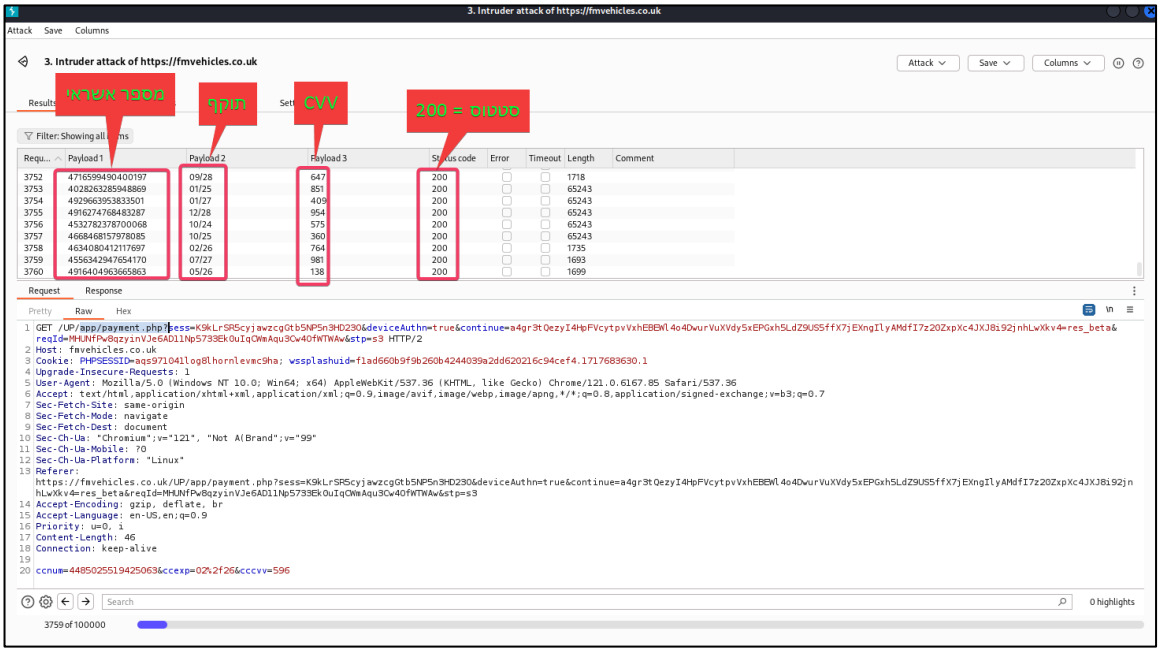
- ניתוב תנועה דרך Tor: על ידי הגדרת Burp Suite לשימוש בפרוקסי SOCKS, אנחנו מבטיחים שכל התעבורה שיורטה ושונתה על ידי Burp Suite מנותבת דרך רשת Tor. זה מספק את יתרונות האנונימיות וסיבוב ה-IP של Tor.
- הימנעות מזיהוי וחסימה: מכיוון שאתר הפישינג עשוי ליישם אמצעים לאיתור וחסימת כתובות IP היוצרות תעבורה חשודה, השימוש ביכולת הרוטציה של כתובות ה-IP של Tor עוזר להימנע מזיהוי וחסימה. נראה שכל בקשה מגיעה מכתובת IP שונה, מה שמקשה על אתר הפישינג לעקוב ולחסום את הפעילויות שלנו.
- שמירה על פרטיות: התחברות דרך Tor מגינה על הפרטיות על ידי מיסוך כתובת ה-IP האמיתית שלנו ושימוש ברשת ה-ONION.

יודגש שאם התוקף השקיע בתשתית הוא יכול לנטר ולגלות איזה בקשות הגיעו מכתובות שהן Tor Exit Nodes וכך לחסום או לאתר את הבקשות הללו. כמובן שניתן להכניס שיפורים לשיטה שלנו על ידי שימוש בטכנולוגיות נוספות, כמו למשל החלפת כתובות באמצעות VPN, רכישת מספר כתובות IP והצפה איטית יותר של נתונים, שילוב של TOR עם כלי אנונימיות נוספים ועוד.

כעת נשאר לנו להכניס את tor אל burp suite, נכנסים להגדרות תחת networks נכנסים אל connections ומגדירים socks proxy:



כעת אנחנו מוכנים לתקיפה, הרצנו את העלאת ה-Payloads הללו:



וניתן לראות שכל התגובות הם 200 כלומר שהתוקף מקבל את המידע הזה. התמונה מציגה את הכלי Burp Suite Intruder בפעולה, שאיתו 'תקפנו' את מערך הפישינג עם סדרה של בקשות HTTP מלאות בנתוני כרטיס אשראי. כאשר נפרק את המרכיבים המרכזיים הנראים ניתן לראות כיצד זה מדגים את ההצפה המוצלחת של מסד הנתונים של התוקף בכרטיסי אשראי.

החלק העליון של התמונה מציג את התוצאות של מתקפת ה-Intruder, מה שמצביע על כך שמספר רב של בקשות ניגשו לאתר הפישינג (fmvehicles[.]co[.]uk).

יש לציין, ניתן לראות שבמהלך השליחה לא שונה ה-user agent של השולח - כך שתוקף יכול להשתמש עם מידע זה כדי לסלק בקשות שיכולות להרוס לו את מערך הפישינג. לכן אם מנסים להציף את התוקף במידע והתוקף עולה על זה, יש לשנות גם את ה-user agent כך שלא יהיה לתוקף דרך לנטרל את הגעת המידע המציף אותו.

ה-Payloads:

העמודות המסומנות "payload 1", "payload 2" ו"payload 3" מתאימות לחלקים השונים של נתוני כרטיס האשראי הנשלחים:

1. מספר כרטיס אשראי
2. תאריך תפוגה
3. CVV

קוד 200: העמודה "status code" מציגה 200 עבור כל בקשה, מה שמציין שהשרת עיבד בהצלחה כל בקשה. המשמעות היא שאתר הפישינג מקבל את נתוני כרטיס האשראי ללא שגיאה. קודי ה-HTTP החוזרים ונשנים שמראים הצלחה (200) מצביעים על כך שכל בקשה עם נתוני כרטיס אשראי מתקבלת על ידי מסד הנתונים של מערך הפישינג. מכיוון ש-Burp Suite מוגדר לשלוח אלפי בקשות אלו, מסד הנתונים מוצף בנתונים שהגדרנו.

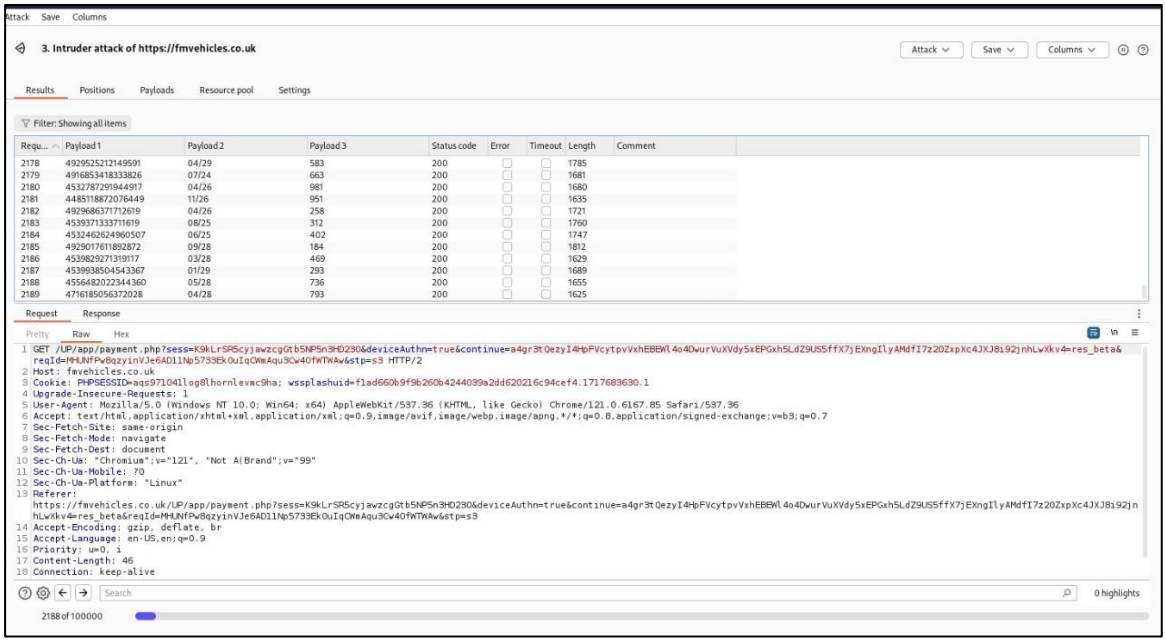
כדי להדגים שהכנסנו בהצלחה נתונים למסד הנתונים של התוקף באמצעות קובץ ה-cookie של ההפעלה, נוכל לבצע את השלבים הבאים:

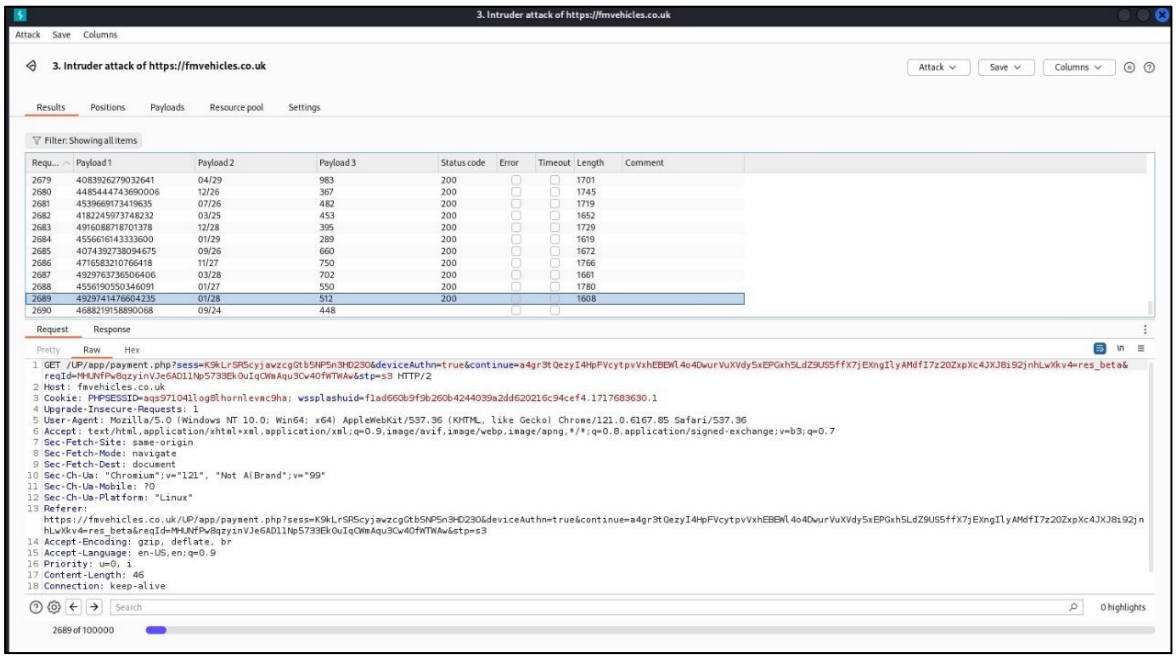
הבנת קובצי ה-Session Cookie: קובץ Cookie (...sess=K9kLrSFS) הוא מזהה ייחודי ששרת האינטרנט מקצה להפעלה של המשתמש. זה עוזר לשרת לעקוב אחר הפעילויות של המשתמש על פני בקשות שונות.

שליחת בקשות עם Burp Suite: בצילום המסך של Burp Suite, כל בקשה לכתובת ההתחזות כוללת את קובץ ה-cookie של ההפעלה. על ידי בחינת תגובות השרת לבקשות אלו, ניתן להסיק שהנתונים ששלחת עובדו על ידי המערכת של התוקף.

מעקב באמצעות קובצי Cookie של session: כל קובץ Cookie של session הוא ייחודי ל-session. אם אנחנו נשתמש באותו קובץ Cookie של הפעלה על פני מספר בקשות, זה מציין לשרת שכל הבקשות הללו מגיעות מאותה session. על ידי מעקב אחר קובץ ה-cookie של ה-session, כך ניתן להבטיח שהשרת שומר על מצב האינטראקציה של המשתמש, מה שמצביע על כך שהשרת עיבד את הנתונים כחלק מהפעלה רציפה.

בתמונות הבאות ניתן לראות שני request שונים [מספר 2188 ומספר 2689] שלשניהם עדיין אותו cookie session מה שאומר שהשרת מעבד את כל הנתונים ולא חוסם את המשתמש באמצעות זיהוי ה-cookie שלו:





הצפה של מסד נתונים עם נתונים חסרי תועלת: על ידי הזרקת כמות גדולה של נתוני כרטיסי אשראי, מה שמתרחש למעשה הוא "הצפה" של מסד הנתונים של התוקף בנתוני אשראי לגיטימיים אך לא שימושיים. זה מקשה מאוד על התוקף להבחין בין הנתונים הגנובים לבין הנתונים הלא אמיתיים, מה שהופך את הנתונים הגנובים לחסרי תועלת.

כמו כן יש לזכור את ההשפעה של הפעולות הללו על תהליך הסליקה, בסופו של דבר התוקף מנסה לאמת כל פרטי אשראי שמוזן על מנת לאמת ב-SMS מחברת האשראי. כלומר שהתוקף מציף את חברת האשראי. ניתן לראות את ה-SMS שמתקבל מחברת האשראי בתמונה הבאה:

ה-SMS הזה הוא בעצם מה שמכונה one time password - OTP. אך ראשית כל יש להבין מה הוא OTP[5]?

OTP מייצג סיסמה חד פעמית. זהו קוד ייחודי וזמני המורכב מארבעה עד שישה מספרים שנוצרו באקראי על ידי הבנק כדי לאמת עסקאות בכרטיס אשראי. המשמעות היא שהעסקה בכרטיס האשראי לא תעבור בהצלחה בלעדיה.

זה משמש כשכבת אבטחה נוספת עבור משתמשי כרטיסי אשראי ומשמש כסוג של התראה עבורם כאשר מישהו מנסה להשתמש בכרטיסי האשראי שלהם ללא רשותו.

[5] התיאור הוא לפי מאמר ב-forbs: https://www.forbes.com/advisor/in/credit-card/what-is-otp-in-a-credit-card-transaction/

איך עובד OTP?

אם מישהו משתמש בכרטיס האשראי שלו כדי לקנות את התיק היקר הזה ממותג צרפתי עם שני C's שלובים על הלוגו. כאשר קונים באינטרנט, אין מי שיבדוק בקצה החנות אם הכרטיס לגיטימי או לא. כאן נכנסת OTP כדי לאמת את הזהות ואת הלגיטימיות של כרטיס האשראי ושל בעלי הכרטיס.

התהליך מתחיל כאשר מזינים את פרטי כרטיס האשראי שלך באתר האינטרנט של החנות. לאחר שכל המידע הדרוש נמסר בטופס ההזמנה, סוחר החנות ישלח בקשה ל-OTP למספר הנייד שנרשם או לכתובת האימייל שניתן.

לאחר שיתקבל ה-OTP, יש להזין אותו באתר כדי להשלים את הרכישה. ה-OTP יגיע מקו ההודעות הרשמי של הבנק, וברגע שיתקבל את ה-OTP והוא יוזן במקומות שסופקו, העסקה תעבור.

מערכת זו אינה מיועדת רק לחנויות מקוונות. OTP משמשים גם בעת ביצוע עסקאות בכרטיס אשראי באופן אישי הדורשות סכום גבוה יותר, כגון קניית מכשירי חשמל ביתיים חדשים או תשלום עבור הכרטיסים הלוך ושוב לניו יורק.

למה בעצם צריכים OTP בעסקאות כרטיסי אשראי?

הצורך ב-OTP הוא כדי שהעסקאות בכרטיסי האשראי יתבצעו בצורה מאובטחת. זה מונע מקרי הונאה שבהם מישהו משתמש בכרטיס אשראי ללא רשות בעליו. זה מוסיף שכבה נוספת של אבטחה נוספת הן למשתמש כרטיס האשראי והן לחנות.

זה עשוי להיות טרחה עבור אנשים מסוימים להזין את ה-OTP שלהם בכל פעם שהם מבצעים עסקה באמצעות כרטיס האשראי שלהם. ובכל זאת, הם צריכים להקריב קורבן קטן כדי להבטיח שהם בטוחים.

ה-OTP הוא אמצעי אבטחה המשמש את הבנקים כדי לקבוע אם אתה, האדם שנרשם בפועל להשתמש בכרטיס האשראי, הוא המשתמש בפועל בכרטיס כרגע.

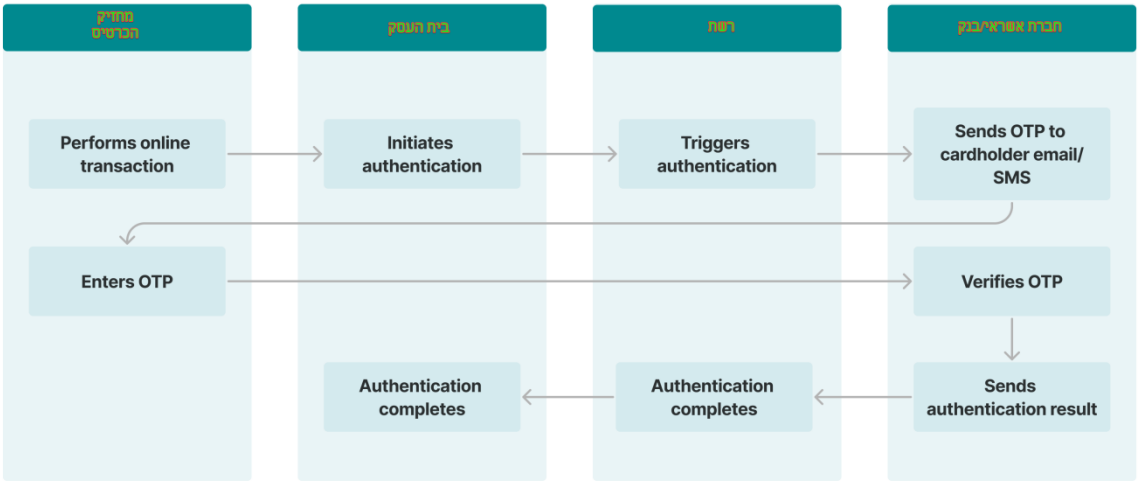
רוב הודעות ה-OTP כוללות אפשרות להתריע לבנק אם מתגלה בקשה לא מורשית ל-OTP, כך שהם יכולים לחסום מיד את העסקה לפני שהיא עוברת.

יתרונות השימוש ב-OTP בעסקאות כרטיסי אשראי

OTPs פועלים כגורם מרתיע עבור גנבים מכיוון שהם יצטרכו לקבל את כרטיס האשראי ואת מספר הטלפון הנייד של בעלי הכרטיס או כתובת הדואר האלקטרוני שלהם כדי להיות מסוגלים ליצור את ה-OTP. זה מקשה עליהם לבצע הונאה. עם OTPs במקום עבור עסקאות כרטיסי אשראי, מקווים למזער את מספר מקרי הונאה.

מלבד הרתעת גנבים, OTPs מציעים גם תחושת ביטחון למשתמשי כרטיסי אשראי. כפי שהוזכר קודם לכן, OTPs הם דרך לבנקים לקבוע אם האדם המשתמש בכרטיס האשראי הוא בעל הכרטיס בפועל. עם מערכת OTP זו, ניתן להבטיח שהכסף שמושג בעמל, יזע ודמעות יהיה בטוח ומאובטח.

בנוסף, OTPs מספקים למשתמש כרטיס האשראי תחושת שליטה. בכל פעם שמתבצע עסקה בכרטיס אשראי, בעלי הכרטיס יהיה זה שיזין את ה-OTP. המשמעות היא שגם אם פרטי כרטיס האשראי דלפו, לא יוכלו להשתמש בהם ללא רשות הבעלים. המחזיק בפרטי האשראי צריכים לבקש מהבעלים באופן אישי את ה-OTP מכיוון שהבנק ישלח אותו באימייל או במספר הטלפון האישי שלהם:



[איור 10: תרשים תהליך הסליקה]

תהליך הסליקה

1. תהליך הסליקה:

- שלב אישור העסקה: כאשר כרטיס אשראי מוזן למערכת, חברת הסליקה שולחת בקשה לאישור העסקה לבנק המנפיק את הכרטיס.
- בדיקת הכרטיס: הבנק/חברת האשראי המנפיק בודק את פרטי הכרטיס (מספר הכרטיס, תוקף, CVV) כדי לוודא שהכרטיס תקין ושיש מספיק יתרה לביצוע העסקה.
- התגובה: אם הפרטים תקינים והיתרה מספיקה, הבנק/חברת האשראי מאשר את העסקה. אם יש בעיות כלשהן (כרטיס לא תקין, פרטים שגויים, חוסר יתרה), הבנק דוחה את העסקה.
- לאחר בדיקת הכרטיס נשלח OTP אל המשתמש, בבקשה שיזין את ה-OTP כפי שמתואר לעיל.
- אם ה-OTP שהוכנס תקין הבנק/חברת האשראי. אם הוכנס בצורה לא נכונה הבנק דוחה את העסקה.

2. הזיהוי על ידי חברת הסליקה:

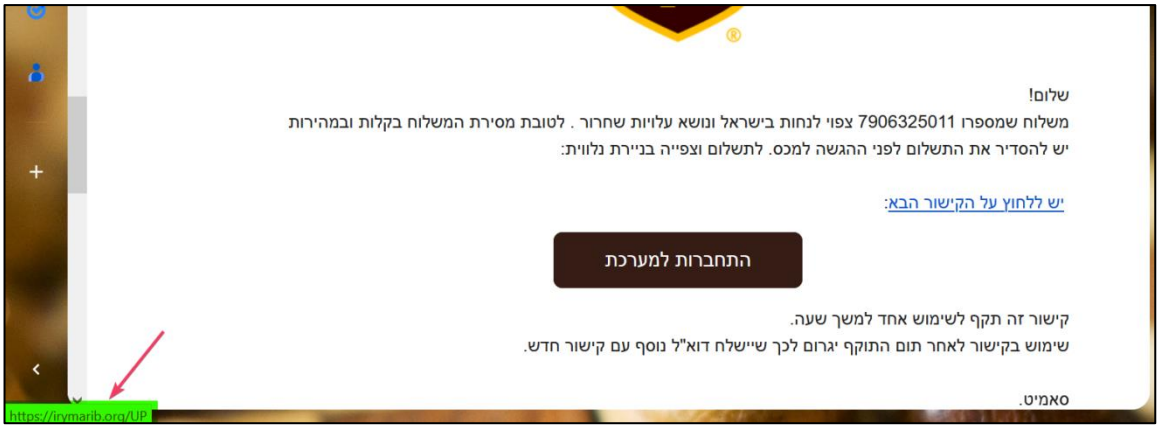
- זיהוי ניסיונות רבים: חברות סליקה משתמשות במערכות לזיהוי התנהגות חריגה. אם יש כמות גדולה של ניסיונות חיוב כושלים מכרטיסים לא תקינים, המערכות יכולות לזהות זאת ולסמן את הפעילות כחשודה.

- חסימת פעילות: במקרים כאלה, חברת הסליקה עשויה לחסום את הניסיונות מהכתובת ה-IP שמהם מגיעות הבקשות, או לנקוט צעדים נוספים כמו בקשת אימות נוסף[6].

כך שגם חברות הסליקה יזהו את העובדה שמתרחשת פעילות חשודה מהאתר, ויתחילו לשאול שאלות על מה שנעשה באתר, ואף לגרום למחזיק האתר לשנות כתובת IP ודומיין כדי שיוכל להמשיך לעבוד.

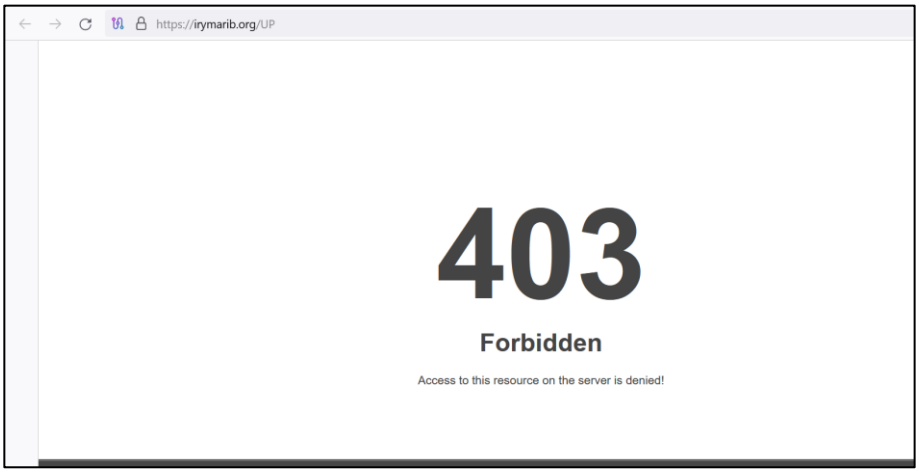
שריפת מערך התקיפה של התוקף

הרצנו את מערך ההגנה שבנינו ולאחר כמה שעות ראינו משהו מעניין מאד, שינוי בפועל של מערך התקיפה של התוקף. הכתובת בקישור לאתר הפישינג היה במקור: https://fmvehicles[.]co[.]uk כפי שראינו, אך לפתע באותו מייל פישינג ניתן לראות כתובת (דומיין) אחרת:



[איור 11: שינוי הדומיין בקישור המרכזי של קמפיין התקיפה]

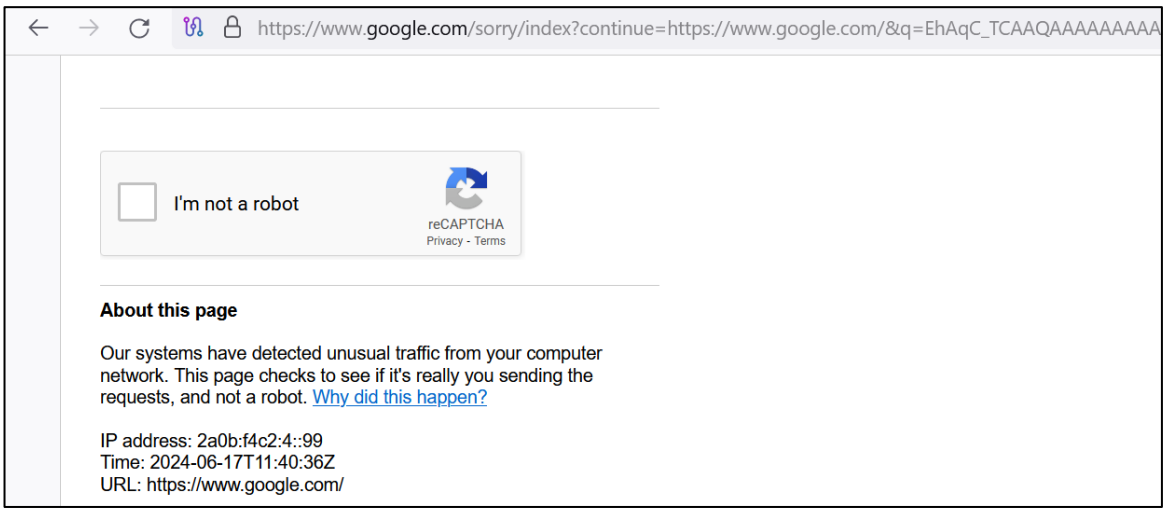
ניתן לראות הפניה לקישור: https://irymarib[.]org/UP קישור זה מחזיר שגיאה 403:



[איור 12: קבלת שגיאה 403 בניסיון לגשת לדומיין החדש]

[6] ראו לדוגמה:
https://finix.com/docs/guides/payments/risk/fraud-detection/
https://www.tinybird.co/blog-posts/how-to-build-a-real-time-fraud-detection-system

ראשית זה מראה על כך שהתוקף שינה את מערך הפישינג שלו, הוא עובר ל-domain אחר. כאשר מנסים להגיע אל הקישור המקורי https://fmvehicles[.]co[.]uk/UP ישנו redirection אל גוגל, יחד עם CAPTCHA:



[איור 13: ניסיון גישה לדומיין המקורי של קמפיין התקיפה (fmvehicles[.]co[.]uk)]

מטורף! הצלחנו בטווח זמן קצר יחסית להשפיע על התוקף ועל מערך התקיפה שלו. לא רק שנכסיו שובשו וכעת הוא לא יכול לגנוב או לחייב כרטיסי אשראי, הוא גם אינו יודע מאיפה המגן (אנחנו) משבשים אותו (לא יודע איך ניתן לעצור את ההצפה) ולכן הוא נאלץ "לשרוף" את כל המיילים ששלח בעבר לקורבנות, כדי שנפסיק להציף אותו. הוא סגר את הכתובת המקורית ואת השרת ששימש אותו לביצוע הגניבה (ניסינו לגשת לכתובת התקיפה המקורית - ללא הצלחה).

יתרה מכך, השפענו גם על המיילים שעתידים להשלח על ידיו, התוקף כעת יצטרך לבחון את רשימת הנמענים שלו כדי לא לשלוח למגן המתוחכם את המייל עם הקמפיין החדש. מעבר לכך, הצפנו את בסיס הנתונים שלו במידע לא ערכי וכך למעשה פגענו ביכולת שלו להשתמש בכרטיסי אשראי הגנובים, למכור אותם לכל המרבה במחיר או להשתמש בהם בעתיד, הוא לא יודע להגיד כמה מהם נכונים ואמיתיים וכמה לא.

חשוב לשים לב לכמה דברים אשר השתנו: דבר ראשון יש להבין כיצד זה קרה שבמייל שאמור להיות סטטי לפתע יש מעבר לקישור אחר, התשובה לכך היא שככל הנראה התוקף עושה שימוש בשירותי שליחת מייל אוטומטיים להרבה לקוחות, שירותים כאלו מאפשרים לשנות תוכן במייל גם לאחר שנשלח, בין השירותים הללו אפשר למנות את Salesforce[7], Simplero[8] ועוד. יש אפילו את mailer שיש בו אפשרות חינמית לשלוח עד 1000 מיילים ואח"כ לשנות בו את התוכן[9].

[7] https://www.youtube.com/watch?v=eU6PHJ-Fq1c
[8] https://www.youtube.com/watch?v=8I9VB55g1Cs
[9] https://www.mailerlite.com/help/how-to-change-a-link-in-a-sent-campaign

ב-(SFMC) Salesforce Marketing Cloud - ואותו הדבר בשאר שירותי שליחת המיילים, ניתן לשנות קישורים בהודעות דוא"ל לאחר שליחת האימיילים עקב האופן שבו SFMC מטפל במעקב אחר קישורים. להלן הסבר מפורט על מנגנון המעקב אחר קליקים וכיצד ניתן לעדכן את הקישורים:

מנגנון מעקב אחר קליקים ב-SFMC עובד בצורה הבאה: כאשר יוצרים אימייל ב-SFMC, זה כולל היפר-קישורים ליעדים שונים. היפר-קישורים אלו אינם מוטמעים ישירות בדוא"ל. במקום זאת, הם הופכים לקישורים שניתנים למעקב על ידי SFMC.

כל היפר-קישור שנמצא במייל מוחלף בכתובת URL שנוצרה על ידי Salesforce, המכונה לעתים קרובות כתובת URL של "link tracker". כתובת ה-URL למעקב אחר קליקים היא כתובת URL ייחודית שנוצרת על ידי SFMC ופועלת כמתווך בין נמען האימייל ליעד הסופי.

כאשר נמען לוחץ על היפר-קישור בדוא"ל, הוא מופנה תחילה לשרת מעקב אחר קליקים של SFMC. שרת מעקב הקליקים רושם את אירוע הקליקים (זמן, משתמש, קישור שנלחץ וכו') למטרות ניתוח ודיווח. לאחר רישום הקליק, השרת מפנה את המשתמש לכתובת אתר היעד בפועל. לכן אפשר לשנות את כתובת אתר היעד לכתובת אחרת, לאחר שמירת השינויים SFMC תפנה כעת את כל הקליקים העתידיים על הקישור הזה לכתובת האתר החדשה[10].

משמעות השינויים במערך התקיפה של התוקף

שיבוש פעולות הפישינג: פעולות פישינג מסתמכות על זרימה קבועה של נתונים חוקיים. על ידי הצפת הדומיין המקורי של התוקף (fmvehicles.co.uk) בנתונים אשראי לא שימושיים, הקשינו עליהם לחלץ מידע שימושי כלשהו וכך גם למנוע את האפשרות למכור את המידע.

החלפת דומיין: הצורך בשינוי דומיינים מעיד על כך שהאתר המקורי של התוקף הפך לפגיע או לא לגיטימי. זה יכול לנבוע מזיהום נתונים, מסד הנתונים מוצף בפרטי כרטיס אשראי ובנוסף חברת הסליקה חסמה את הדומיין בעקבות ריבוי בקשות לאימות התשלום.

השפעת השיבוש על מערך הפישינג של התוקף

חשוב לציין, לא ניצלנו אף חולשה או CVE כזה או אחר, כל פעולה שלנו הייתה חוקית. למעשה השתמשנו ביכולת לגיטימית (פשוט הזנו נתונים כפי שהתבקשנו ☺), כדי לשבש את קמפיין התקיפה של התוקף.

[10] https://help.salesforce.com/s/articleView?id=sf.mc_es_link_tracking.htm&type=5

כאשר אנחנו משתמשים באמצעים יחסית דלים כדי לשבש את מערך הפישינג של התוקף, יש לכך השלכות משמעותיות שמקשות עליו להמשיך בפעילותו:

1. **שריפת מערך המיילים**:

- **זיהוי המיילים**: לאחר שנחשפנו להתקפה והתחלנו להציף את התוקף בכרטיסי אשראי לא שימושיים, כל המיילים שנשלחו על ידי התוקף לנמענים "נשרפים". התוקף לא יכול לדעת מי מבין הנמענים הוא זה שהציף אותו במידע שגוי.
- **ביטול המיילים**: כתוצאה מכך, התוקף צריך לבטל את כל המיילים ששלח או לחילופין לשנות את הלינק לדומיין ששימש לתקיפה, כי הוא לא יכול לסמוך על המידע שהתקבל מהם. זה מחייב אותו להתחיל את התהליך מחדש ולשלוח מיילים חדשים לנמענים חדשים או לנמענים קיימים מסוימים, בתקווה שלא יתגלו שוב.

2. **שינוי מערך התקיפה (דומיינים ושרתים)**:

- **שריפת הקמפיין**: התוקף כבר הבין שהקמפיין הנוכחי (UPS) נשרף והוא צריך להקים קמפיין חדש מההתחלה.
- **שריפת הדומיין**: כאשר אנחנו חושפים את דומיין הפישינג של התוקף (במקרה שלנו fmvehicles[.]co[.]uk) אנחנו גורמים שידעו עליו בחברות הסליקה, אצל ספקי האינטרנט וכדומה. כתוצאה מכך, הדומיין הזה נחשב ל'שרוף', והמשך התקיפה דרך הדומיין הזה נחסמת.
- **שריפת השרתים**: בנוסף לדומיינים, גם כתובות ה IP-של השרתים שבהם משתמש התוקף כדי לארח את אתרי הפישינג נחשפים. כתוצאה מכך, השרתים האלה נחשבים לשרופים ונחסמים על ידי ספקי השירותים השונים.

3. **הקמה מחדש של מערך התקיפה**:

- **החלפת דומיינים ושרתים**: התוקף נאלץ למצוא דומיינים ושרתים חדשים לצורך התקיפות שלו. זהו תהליך שדורש זמן ומשאבים, ומקשה עליו לשמור על יציבות בפעילות הפישינג.
- **התחלת תהליך מחדש**: גם אם התוקף מצליח לשנות את הדומיינים והשרתים, הוא צריך להתחיל את כל תהליך הפישינג מחדש - בנייה מחדש של תבנית המייל, איסוף נמענים חדשים, שליחת מיילים לנמענים, הכנת אתרי פישינג חדשים ושימור התהליך כך שלא יתגלה שוב.

4. **השפעה על האמינות של התוקף**:

- **פגיעה במוניטין**: ככל שהתוקף נתקל ביותר שיבושים וחשיפות, כך האמינות שלו יורדת בעיני שותפיו ולקוחותיו, אם יש לו כאלה. הוא מתקשה לשמור על מוניטין אמין ויעיל בשוק השחור.
- **פחד מחשיפה ואיתור**: התוקף הופך לחשוף יותר לפגיעות משפטיות ואכיפה, כאשר רשויות החוק וספקי השירותים מודעים לפעילותו. הוא צריך להיות יותר זהיר ומודע לסיכונים, מה שמקטין את יעילותו ויכולת הפעולה שלו.

סיכום

באמצעות הצפת התוקף במידע לא שימושי וחשיפת הדומיינים והשרתים שלו, אנחנו מצליחים לשבש בצורה משמעותית את מערך התקיפה ואת קמפיין הפישינג שלו. הוא נאלץ להשקיע זמן ומשאבים רבים בהקמת המערך מחדש, שינוי הקמפיין, החלפת דומיינים ושרתים, ובניית אמינות מחודשת. כל זה מקשה עליו להמשיך בפעילותו ומפחית את הסיכון לקורבנות נוספים. יתרה מכך, הצלחנו להזיז את הגבינה, כל השפעה על התוקף יכולה לגרור אותו לבצע טעויות נוספות שבסופו של יום יובילו לאיתור שלו.

ניתן כמובן לקחת את הטכניקה כמה צעדים קדימה, למשל על ידי שימוש בכרטיסי אשראי המשמשים כמלכודות, כאשר בחיוב שלהם ניתן לעקוב אחר פעילות התוקף, להבין באיזה עסקים מבצע חיובים, לעקוב אחר המיקומים של החיובים עד לאיתור מיקום מדויק ותפיסת התוקף או משתפי הפעולה. במקרים אחרים ניתן לספק כרטיסי אשראי עם מסגרת תקציבית מסוימת, אשר מנטרת מי משתמש בתקציב וכך עולה על עבריינים נוספים, אשר משתמשים בכרטיס לאורך זמן, בסכומים משתנים וכך מועדים לטעויות. לאחר האיתור ניתן יהיה למשוך את הכסף "השקוף" וכך אף ניתן להמנע מהפסד ולמחזר את הכסף עבור קמפיינים נוספים.

על המחברים

יהודה כהן הוא האקר ובודק חוסן מוביל בעל ידע נרחב בתחום הסייבר. עם רקע חזק ורזומה מוכח, הוא הפך לשחקן מרכזי בתעשייה, הידוע בזכות הידע העמוק והבנתו הטכנית.

שי נחום, זוכה "פרס ביטחון ישראל", בוגר יחידת העילית הטכנולוגית - ממר"ם. חוקר סייבר, בודק חוסן תשתיתי ואפליקטיבי, מומחה פורנזיקה וניתוח פוגענים, בעל ניסיון של מעל 15 שנים בתחום הסייבר ואבטחת המידע. בעל רזומה עשיר בתקיפות וזיהוי מתקפות סייבר וחדירה לארגונים בגופים אזרחיים, ממשלתיים וביטחוניים. בוגר הטכניון לתואר שני (M.Sc) בהנדסת מערכות מידע, בהתמחות סייבר עם פיתוח מחקר פורץ דרך בתחום. תואר ראשון בהנדסת מערכות מידע בבן גוריון, בהתמחות סייבר. מעביר הרצאות בתחום הסייבר בכנסים מובילים בארץ ובעולם.