The Perfect Cover Masking Password Sprays as Microsoft Traffic

מאת מתן בכר

הקדמה

לכל אחד ואחת מאיתנו הייתה פעילות שבה התבקשנו להיות כמה שיותר שקטים ולא לחשוף את עצמנו בעת תהליך התקיפה. במחקר שאציג בפניכם, אראה טכניקה שבה ניתן לנצל משאבים בעולמות הענן לטובת תקיפה, הסתרת הזהות האמיתית שלנו (באמצעות יציאה מכתובת IP שונה) ועקיפת מנגנוני הגנה. כל זה באמצעות שירותים הקיימים לנו בסביבת הענן של Azure.

מהם אותם משאבים (Resources) שנוכל לנצל ואיך הם עובדים מאחורי הקלעים?

בעולמות הענן ישנם המון משאבים שניתן לנצל את הפונקצינליות הבסיסית שלהם לטובת מתקפות. במאמר זה נעבור על ניצול של Automation Accounts , Azure RunBook ו- Azure Functions, כמובן שישנם עוד משאבים שניתנים לניצול, כגון Logic Apps שלא נסקור במאמר זה.

Automation Accounts מספק שירות אוטומוציה מבוסס ענן, ניהול עדכוני מערכות הפעלה, מה שמאפשר ניהול עקבי בסביבות Azure וגם בסביבות שאינן Azure.

השירות כולל אוטומציות לתהליכים, ניהול עדכונים, יכולות משותפות ותמיכה בסביבות הטרוגניות.

Automation Accounts מאפשרים בידוד של משאבי האוטומציה, Runbooks, נכסים ותצורות ממשאבים של חשבונות אחרים. ניתן להשתמש ב-Automation Accounts כדי להפריד משאבים לסביבות לוגיות נפרדות או לפי תחומי אחריות מואצלים. לדוגמה, ניתן להשתמש בחשבון אחד לסביבת פיתוח, בחשבון אחר לסביבת ייצור, ובחשבון נוסף לניהול סביבות מקומיות (On-Premises). לחלופין, ניתן להקצות Automation Accounts ייעודי לניהול עדכוני מערכות הפעלה עבור כלל המכונות בארגון באמצעות מנגנון ניהול העדכונים.

Runbooks של Azure Automation מהווים כלי עוצמתי לאוטומציה של משימות שגרתיות ולניהול משאבים ב-Azure, באמצעות שימוש בשפות תכנות כמו PowerShell ו-Python משתמשים יכולים ליצור Runbooks לטובת ייעול תהליכים תפעוליים וביצוע אוטומציות בתשתית הענן.

Azure Functions הוא שירות Serverless המסופק על ידי Microsoft Azure המספק את היכולת להריץ אפליקציות או פונקציות.

ניתן להפעיל את ה-Functions באמצעות מגוון Triggers, כגון בקשות HTTP, Triggers מבוססי זמן, Queues ואירועים המתקבלים משירותים אחרים של Azure, מה שמספק גמישות גבוהה בפיתוח פתרונות אוטומוציה, אינטגרציה ועיבוד אירועים בענן.

כאשר מתבצעת הרצה ל-Function מאחורי הקלעים ישנו 3-way handshake ולאחר מכן נוצר לנו חיבור מאובטח שאותו נוכל לראות בתמונה הבאה:

No.	Time	Source	Destination	Protocol	Lengtl	Info
15	2.395179	[redacted]	20.48.204.6	TCP	66	51431 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
16	2.542340	20.48.204.6	[redacted]	TCP	66	443 → 51431 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM
17	2.542411	[redacted]	20.48.204.6	TCP	54	51431 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=0
18	2.545220	[redacted]	20.48.204.6	TLSv1.3	512	Client Hello (SNI=[redacted].azurewebsites.net)
19	2.692889	20.48.204.6	[redacted]	TLSv1.3	1506	Server Hello, Change Cipher Spec
20	2.692889	20.48.204.6	[redacted]	TCP	1506	443 → 51431 [ACK] Seq=1453 Ack=459 Win=4194304 Len=1452 [TCP PDU reassembled in 22]
21	2.692889	20.48.204.6	[redacted]	TCP	1506	443 → 51431 [ACK] Seq=2905 Ack=459 Win=4194304 Len=1452 [TCP PDU reassembled in 22]
22	2.692889	20.48.204.6	[redacted]	TLSv1.3	214	Application Data
23	2.692968	[redacted]	20.48.204.6	TCP	54	51431 → 443 [ACK] Seq=459 Ack=4517 Win=65280 Len=0
24	2.696807	[redacted]	20.48.204.6	TLSv1.3	407	Change Cipher Spec, Application Data, Application Data
25	2.840125	20.48.204.6	[redacted]	TLSv1.3	157	Application Data
26	2.894592	[redacted]	20.48.204.6	TCP	54	51431 → 443 [ACK] Seq=812 Ack=4620 Win=65280 Len=0
122	7.517669	20.48.204.6	[redacted]	TLSv1.3	330	Application Data
123	7.517669	20.48.204.6	[redacted]	TLSv1.3	81	Application Data
124	7.517731	[redacted]	20.48.204.6	TCP	54	51431 → 443 [ACK] Seq=812 Ack=4923 Win=65024 Len=0

כאשר המחשב שלנו מבצע חיבור בצורה מאובטחת אל מול ה-Azure Function App, תחילה מתבצע חיבור בסיסי באמצעות TCP Handshake.

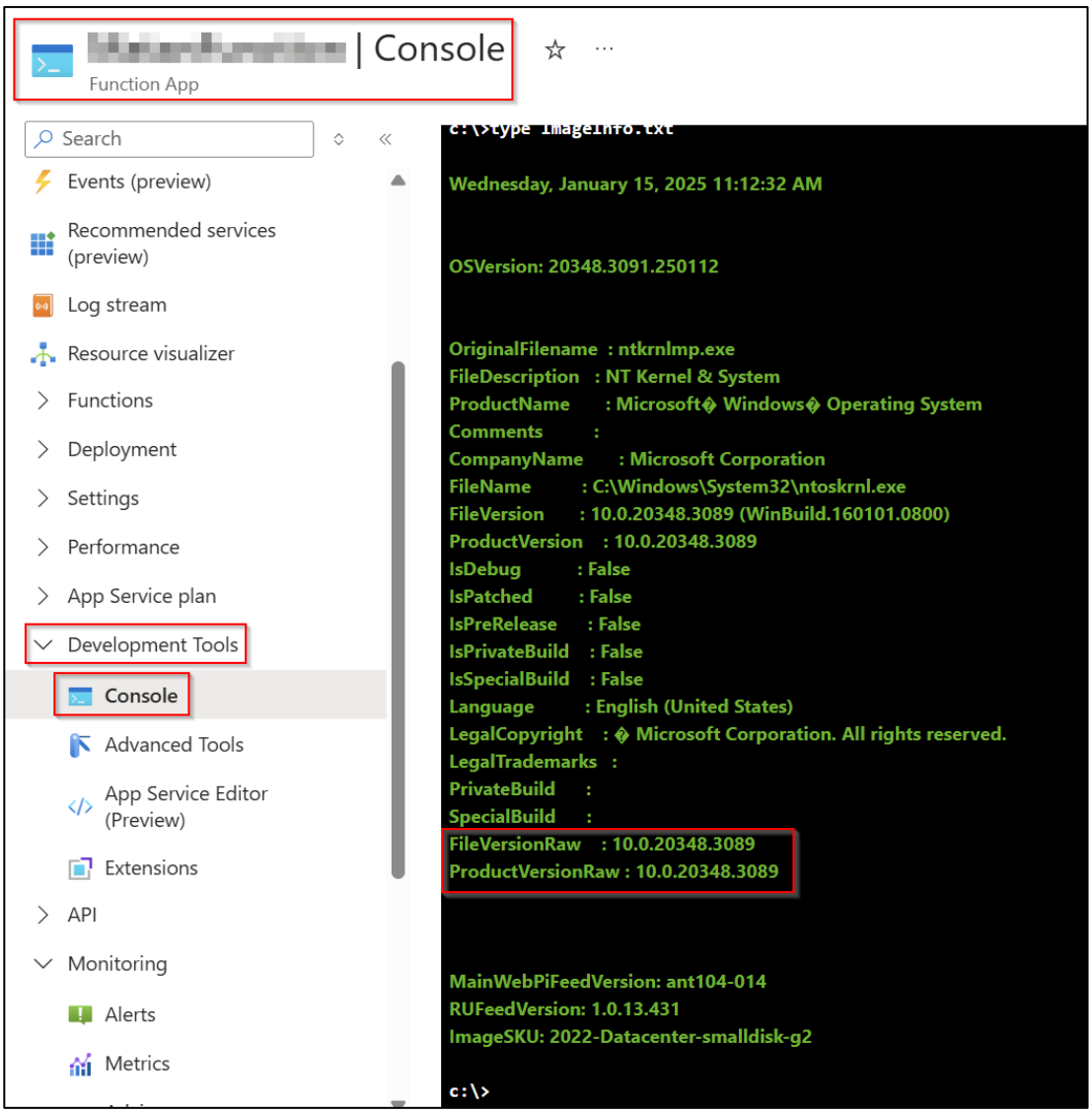
מיד לאחר מכן מתבצע שימוש בערוץ מאובטח על ידי שימוש ב-HTTPS עם TLS v1.3 (שהינו שיטת ההצפנה החדשה והמאובטחת ביותר שקיימת היום), מה שמאפשר לנתונים שלנו כמו סיסמאות או מידע רגיש אחר להיות מוצפנים.

לאחר שההצפנה קיימת, ההעברת נתונים בין התחנה שלנו לבין Azure מתרחשת באופן מאובטח ומופיע כ-Application Data מוצפן.

כיצד Azure Function עובד מאחורי הקלעים?

כאשר אנו מבצעים Deploy לקוד ב-Function App, Azure מבצעים Provision למשאבים בצורה דינאמית על התשתית של מיקרוסופט, מאחורי הקלעים.

בתמונה הבאה נוכל לראות את השרת וה-Instance שמריץ את הפונקציה שלנו:



בתמונה מעל נוכל לראות את ה-"Kudo Console" אשר הינו ממשק הניהול של השרת שמאחסן ומריץ את הפונקציה שלנו ב-Azure.

השרת שמופיע בתמונה למעלה הינו Windows Server 2022 Build 20348.

ה-Function apps מסתמכות על מגוון שירותי Azure לטובת הפעולה שלהן.

תמיד יהיה Storage account שמשוייך ל-Function App לטובת ניהול ה-Triggers והלוגים.

Triggers ו-Bindings מספקים שכבה לאינטגרציה עם שירותים אחרים, לדוגמא טריגר של Event Hub משתמש בשירות של Azure Event Hub וכו'.

כאשר Event מתרחש, ה-**Scale Controller** של Azure Function (רכיב ב-Azure) מזהה אותו ומקצה משאבי מחשוב בהתאם. קוד הפונקציה נטען ומורץ, ובמהלך ההרצה הוא עשוי לקרוא ל-API-ים חיצוניים, לגשת לבסיסי נתונים או לבצע כל פעולה שהוגדרה בקוד.

לאחר סיום ההרצה, התוצאות או פלטים ישלחו ליעד שנבחר, לדוגמא כתיבה לבסיס נתונים.

מאחר שהפלטפורמה מבצעת סקיילינג אוטומטי, מספר instances של אותה פונקציה יכולים לרוץ במקביל במקרה של עומס אירועים גבוה.

כל התהליך מנוהל על ידי Azure App Service מה שאומר ש-Azure אחראית על עדכוני מערכת ההפעלה של ה-VM-ים הבסיסיים, ניהול רשת, קונפיגורציות אבטחה ותחזוקה שוטפת.

למשתמש אין גישה ישירה למערכת ההפעלה של ה-Host או למערכת הקבצים, מעבר לקוד שנפרס (Deployed).

מודל ההרצה המנוהל הזה נוח מאוד לשימוש אך מחייב אותנו לסמוך על מנגנוני האבטחה של Azure שכן כל חולשה בסביבת ה-Sandbox או בפלטפורמה עצמה עלולה בצורה תאורטית כמובן להיות מנוצלת על ידי שימוש בקוד זדוני.

אז מה קורה בעת הטריגר של ה-Function?

כאשר מתבצע טריגר ל-Azure Function ישנם מספרים דברים שקוראים באופן מיידי.

1. **Trigger Event** – Azure מאזינים באופן רציף ל-Events שהגדרנו כגון בקשת Web, Timer Trigger או אפילו עדכון של הבסיס נתונים.
2. **Runtime Preparation (Cold Start)** – תחילה Azure מקצים סביבה מבודדת ומאובטחת (Sandbox) ל-Function, לאחר מכן הקוד של הפונקציה נטען, מתקין את הספריות הנדרשות ומכין את הסביבה לטובת ההרצה (כמו Python, Node.js או .NET).
3. **Code Execution** – הפונקציה מורצת בתוך ה-Sandbox, הסביבה מספקת הכל לקוד כדי שירוץ בצורה חלקה, כולל משאבי מחשוב, אחסון זמני וחיבור תקשורתי.
4. **Binding and Outputs** – Azure בצורה אוטומטית מחברים את הפונקציה עם הקלטים והפלטים ההכרחיים, כמו בסיסי נתונים או Message Queues, דרך מה שידוע כ-Bindings. לאחר סיום ההרצה של הפונקציה, Azure מעבירים בצורה חלקה את הפלט לאן שהוא אמר להגיע.
5. **Cleanup or Reuse** – לאחר ההרצה, Azure עלולים להשאיר את הסביבה בתצורה "חמה" על מנת להיות מוכנים לבקשה הבאה מה שמאיץ משמעותית את ההרצות הבאות.

כעת נבצע מעבר על ההבדלים בין Cold Start ל-Warm Start:

Cold Start – מתרחש כאשר אנו מריצים את הפונקציה בפעם הראשונה (או שלא הרצנו אותה למשך זמן רב). Azure תחילה צריך ליצור את הסביבה, לייבא את הקוד ולהתקין את הספריות הנדרשות, ההתקנה הראשונית הזו יכולה לקחת מעט זמן.

Warm Start – הרבה יותר מהירה, לאחר ההרצה הראשונה, Azure משאירים את הסביבה מורצת בציפייה ל-Triggers חדשים, כך שבעת שיקרו יוכלו להשתמש בסביבה זו מה שיאפשר הרצה זריזה יותר.

Azure Functions מבצעים סקיילינג אוטומטי בהתאם לעומס העבודה:

- כאשר מתקבלת כמות גדולה של בקשות או אירועים, Azure מקצה באופן מיידי משאבי מחשוב נוספים.
- כאשר רמת הפעילות יורדת, הפלטפורמה מבצעת הקטנה הדרגתית של המשאבים ובכך חוסכת בעלויות.

גמישות זו היא אחד הגורמים המרכזיים שהופכים את Azure Function לפתרון עוצמתי, יעיל וחסכוני במיוחד מבחינת העלות.

Security and Isolation – Azure Functions פועלות בתוך קונטיינירים מאובטחים ומבודדים (Sandboxed Containers).

כל פונקציה של לקוח מורצת בסביבה נפרדת.

Azure מנהלים את הקונטיינרים ומיישמים עדכוני אבטחה ותיקונים (Patching) כך שהמפתחים אינם נדרשים לטפל בכך בעצמם.

Monitoring and Logging – Azure Functions מגיעות עם אינטגרציה מובנית ל-Application Insights מה שמאפשר לקבל תובנות מעמיקות לגבי הרצה, ביצועים ותקלות אפשריות של הפונקציות.

מאחורי הקלעים Azure אוספים באופן אוטומטי לוגים של הרצה, מדדים (Metrics) ושגיאות ומספקים למפתחים יכולות אבחון (Diagnostics) קריטיות לניטור ופתרון תקלות.

למה הכתובת IP של מיקרוסופט תהיה בלוגים של הקורבן?

מכיוון ש-Function Apps רצות על התשתית המנוהלת של Azure, התקשורת שיוצאת החוצה נעשית על ידי הפונקציה יוצאת מכתובת IP של מרכזי הנתונים של Microsoft, כתוצאה מכך, כאשר נשלחות בקשות או מתבצעות בקשות לשירותים חיצוניים, הלוגים בצד של היעד (לוגים של הקורבן אותו נתקוף) יתעדו את כתובת ה-IP הציבורית של Azure ולא את הכתובת IP של התוקף.

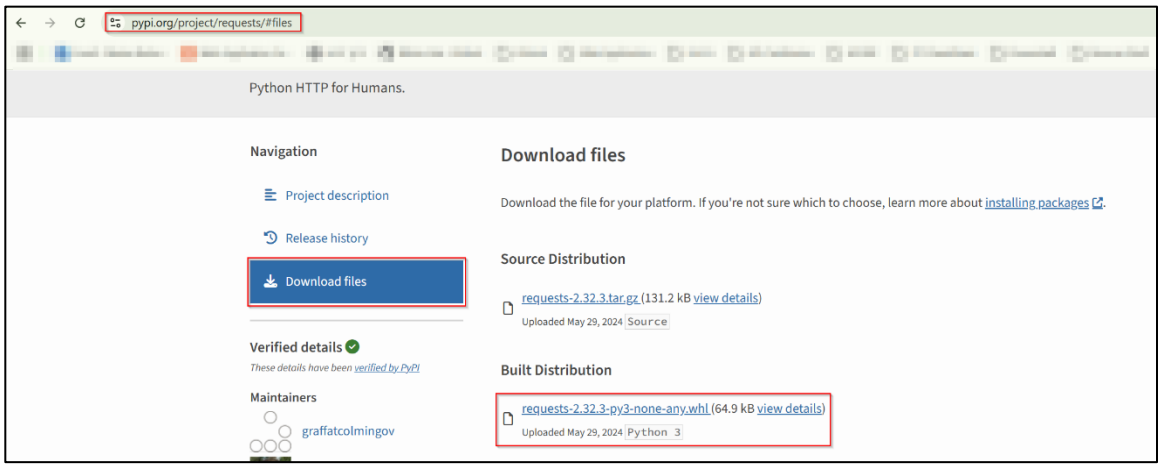
תרחיש תקיפה – Automation Accounts

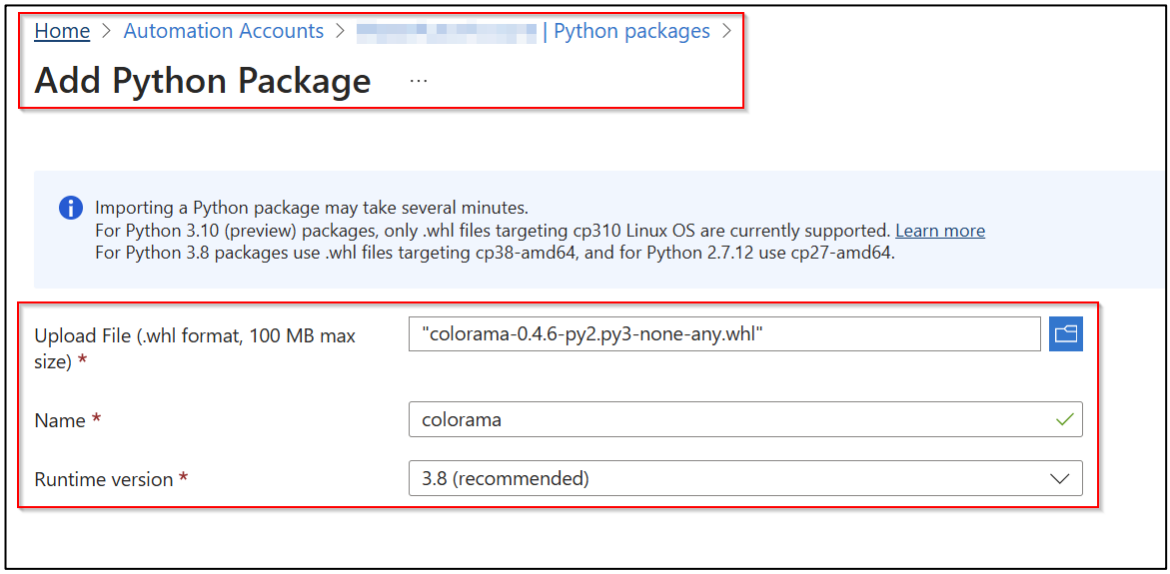
לאחר שעברנו על המשאבים הגיע הזמן להראות איך ניתן לנצל אותם בתרחיש תקיפה.

יצא לנו להשתמש ב-Password Spray בלא מעט פעילויות, אבל כחלק מהסיכונים הקיימים כאשר מבצעים את המתקפה בעולמות הענן, כתובת ה-IP בה אנו משתמשים תופיע בלוגים של הקורבן. בנוסף, יכולות להיות מופעלות Conditional Access Policies אשר יגבילו את הגישה שלנו, כך שלעיתים שימוש בתשתית Microsoft ויציאה מכתובת IP שנראית לגיטימית יכולים לעזור לנו להתגבר על כך.

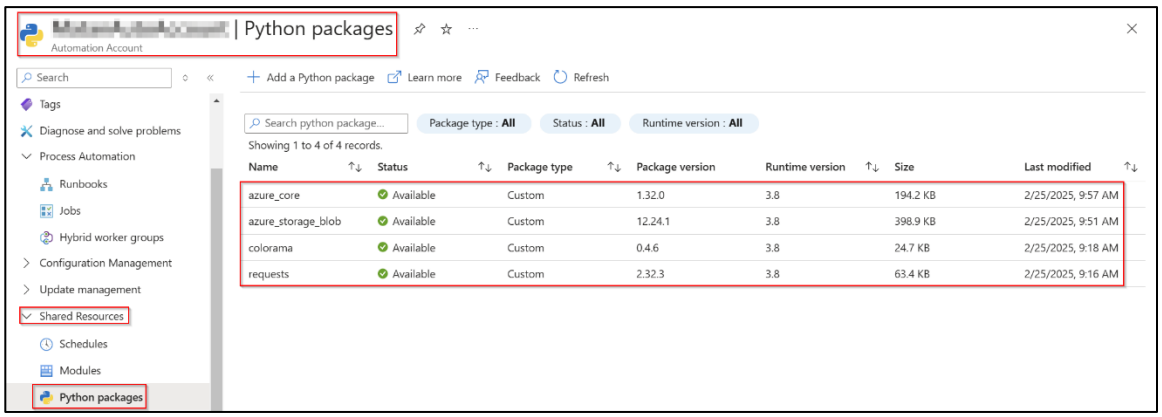
תחילה, לטובת תרחיש התקיפה ניצור Automation Accounts ונוסיף את הסקריפט Python שלנו ל-Runbook.

לאחר מכן נבצע Import ל-Packages שנצטרך לטובת תרחיש התקיפה.





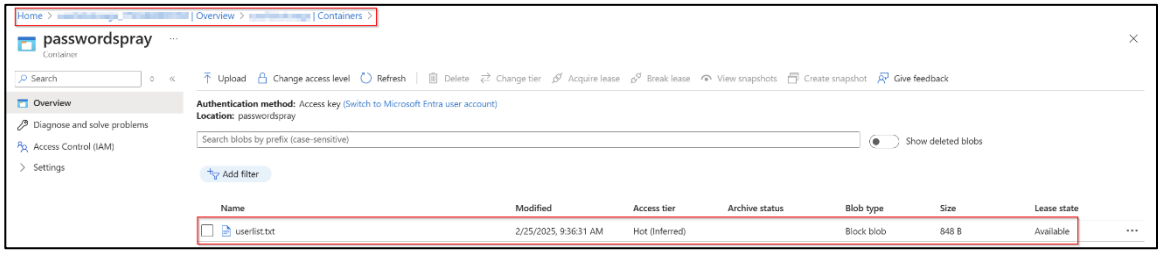
בתמונה הבאה נוכל לראות את כל ה-Packages שרלוונטים לטובת ההרצה של ה-Runbook.



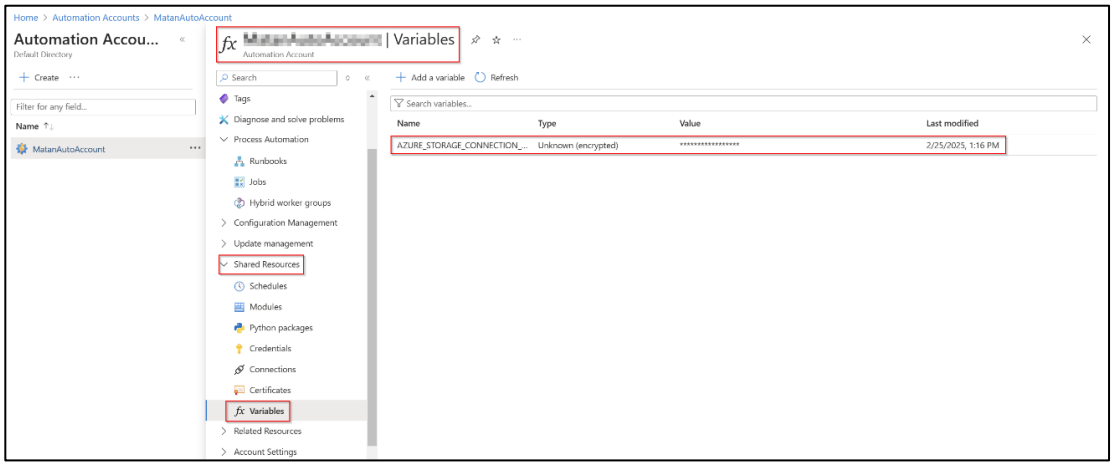
הצעד הבא שלנו זה לעלות רשימת משתמשים ל-Storage Account.

Storage Account ב-Azure הינו פתרון אחסון מבוסס ענן המספק אחסון סקיילבילי, מאובטח ובעל זמינות גבוהה עבור סוגי נתונים שונים.

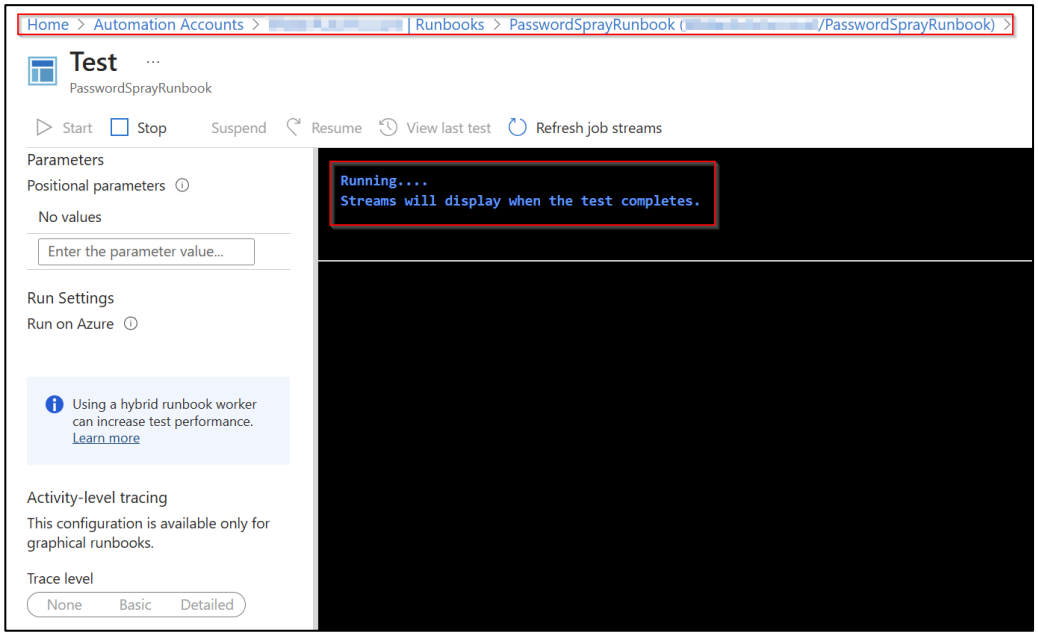
השירות תומך במספר שירותי אחסון, ביניהם Blob Storage (לאחסון נתונים כגון תמונות, סרטונים וכו'), File Shares (לאחסון קבצים מבוסס SMB) וטבלאות (לנתוני NoSQL).

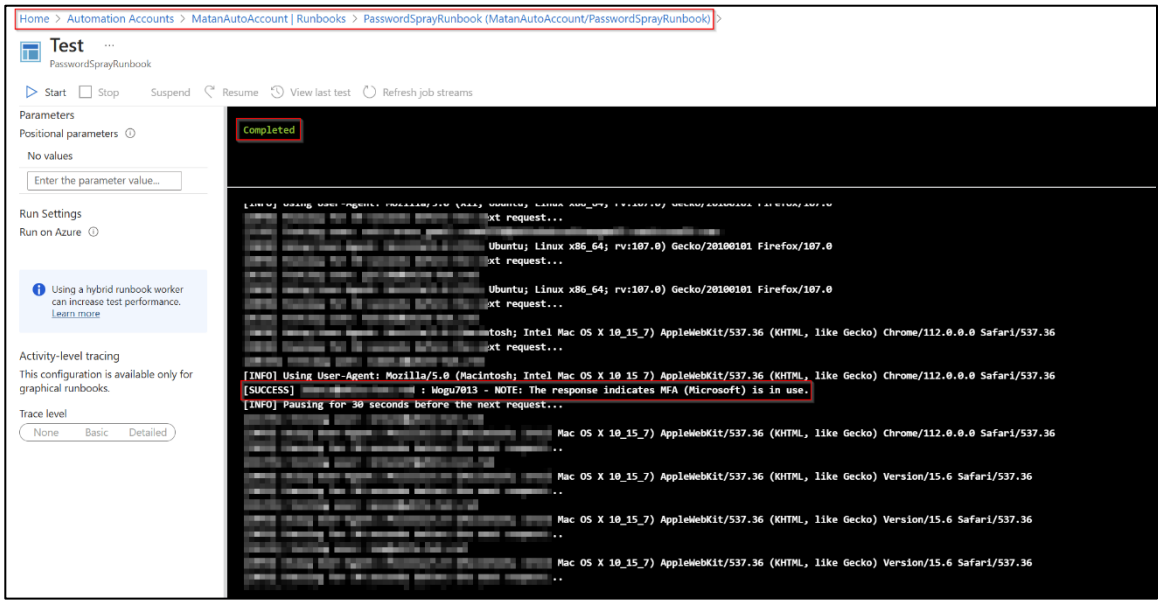


לאחר מכן נוסיף Connection String (לטובת התממשקות אל מול ה-Storage Account שיצרנו) ל-Environment Variable (אופציונלי).

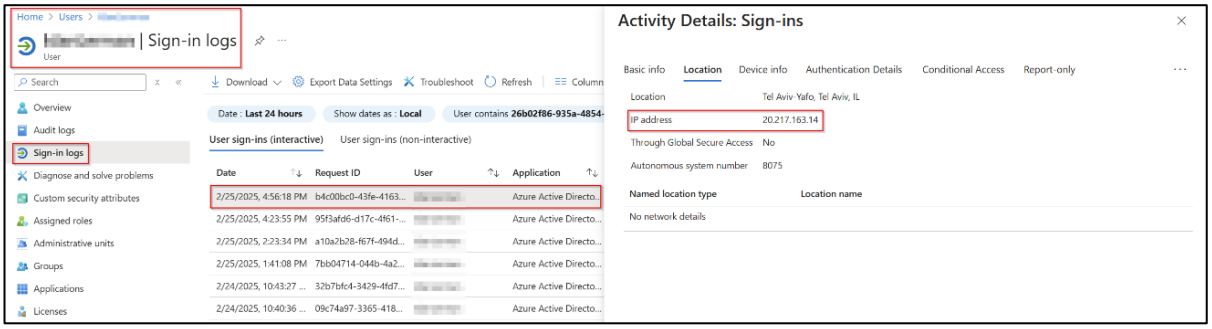


כעת נבצע הרצה ל-Runbook.





לאחר שתרחיש התקיפה הסתיים נוכל לראות בלוגים 2 כתובות IP שונות אשר שייכות ל-Microsoft.



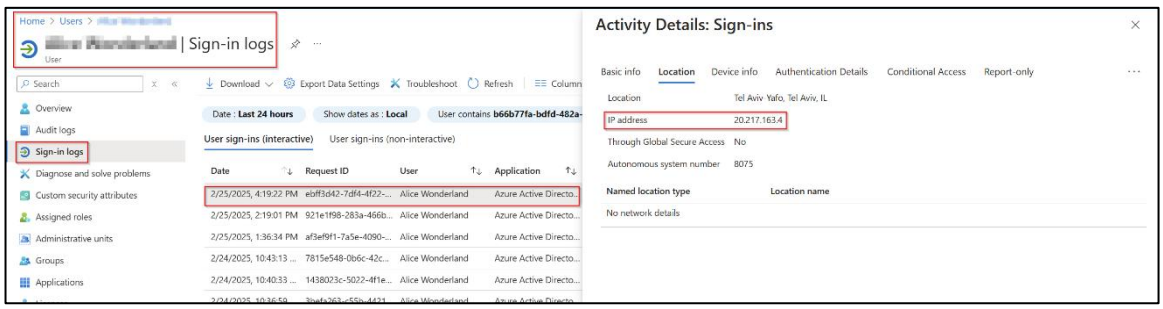
20.217.163.14 address profile

Whois | Diagnostics

IP Whois

```
NetRange:       20.192.0.0 - 20.255.255.255
CIDR:           20.192.0.0/10
NetName:        MSFT
NetHandle:      NET-20-192-0-0-1
Parent:         NET20 (NET-20-0-0-0-0)
NetType:        Direct Allocation
OriginAS:
Organization:   Microsoft Corporation (MSFT)
RegDate:        2017-10-18
Updated:        2021-12-14
Ref:            https://rdap.arin.net/registry/ip/20.192.0.0


OrgName:        Microsoft Corporation
OrgId:          MSFT
Address:        One Microsoft Way
City:           Redmond
StateProv:      WA
PostalCode:     98052
Country:        US
RegDate:        1998-07-10
Updated:        2024-03-18
```



חשוב להבין ש-Azure Runbook הוא חלק מ-Azure Automation, זאת אומרת, הוא מריץ משימות על ידי שימוש בתשתית הענן של Micrososft, והכתובות IP שמהן אנו יוצאים שונות בכל Region.

כל Azure Region מקצה טווח יחודי של כתובות IP ל-Automation Accountss מה שאומר שאם ה-Runbook יורץ מ-Regions שונים אז הכתובת IP שממנה ניצא תהיה שונה גם כן.

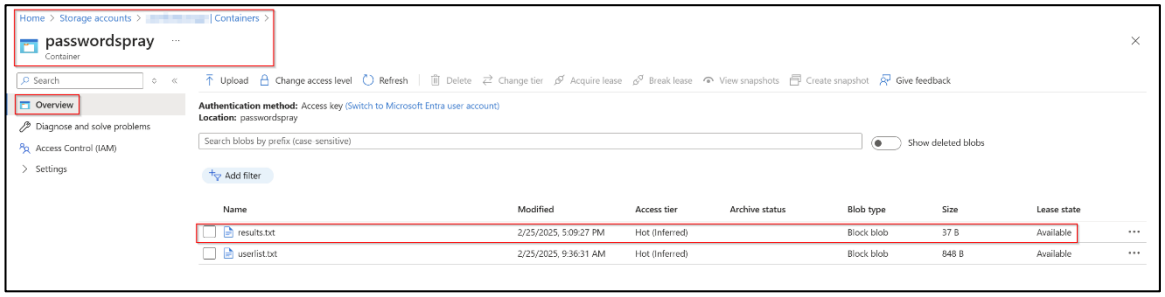
כתוקפים נוכל ליצור מספר Automation Accountss ב-Regions שונים ונשתמש ב-Runbooks בכל Automation Accounts כדי להריץ את המתקפות הזדוניות שלנו, כך שהצד השני (הקורבן) יראה כתובת IP לגיטימית כשנבצע Password Spray או DOS או כל מתקפה אחרת.

יהיה קשה לבצע חסימה למתקפות אלו מכיוון שהן יוצאות ומגיעות מכתובת IP של Microsoft.

למעבר על כלל הכתובות הציבוריות של כל Region:

[Download Azure IP Ranges and Service Tags – Public Cloud from Official Microsoft Download Center]

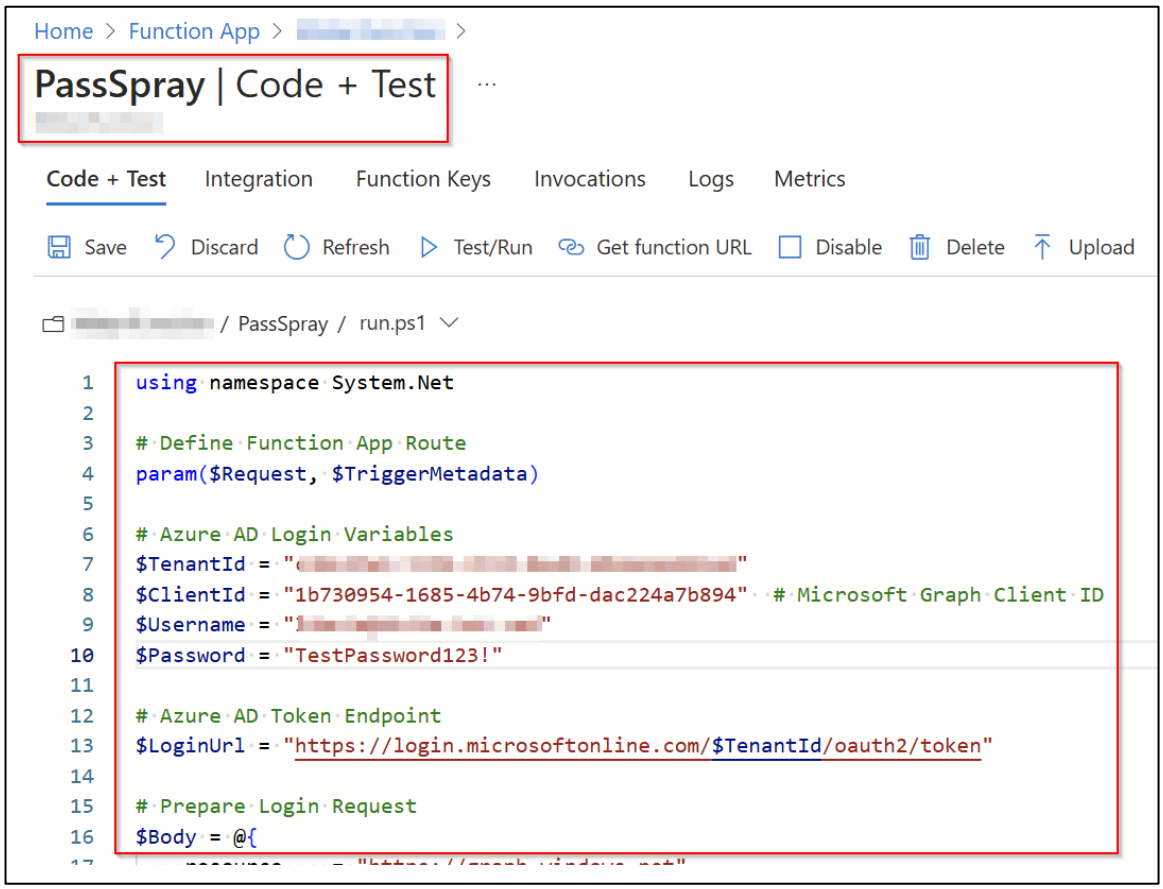
לאחר סיום תהליך התקיפה נוכל לראות את הקובץ עם המשתמשים שהשגנו:



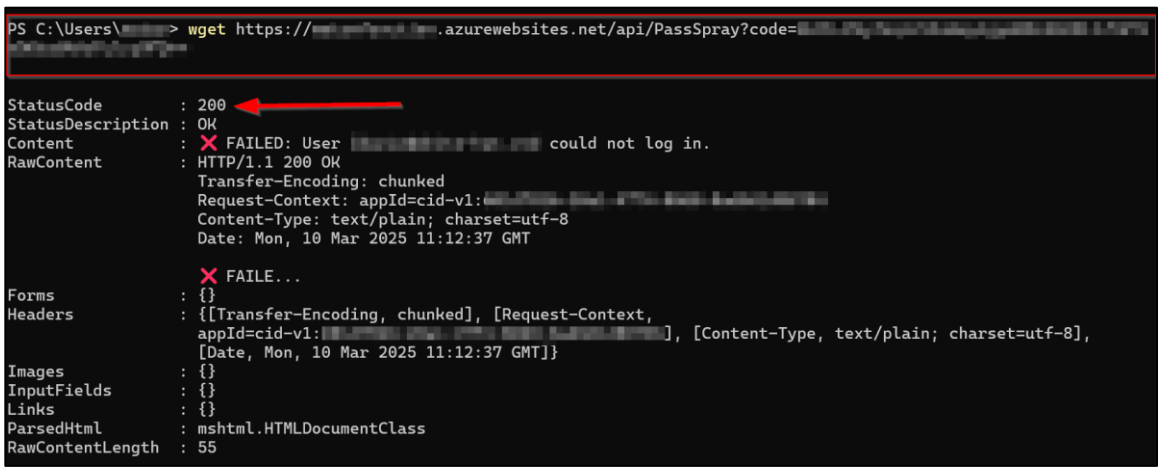
תרחיש תקיפה – Function App

ישנם לא מעט דרכים לבצע את תהליך התקיפה שהדגמתי קודם לכן עם ה-Automation Accounts, כעת אציג דרך נוספת באמצעות Function App.

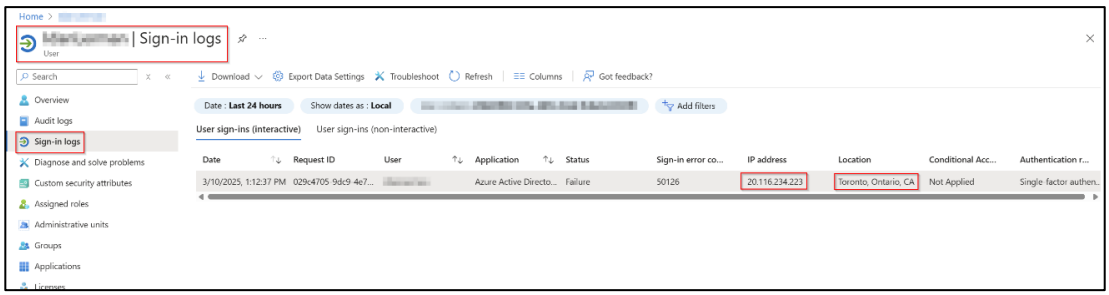
תחילה ניצור Function App וניצור פונקציה להרצת הקוד (אני השתמשתי ב-HTTP Trigger וב- PowerShell Core לטובת ההרצת קוד):



כעת נבצע את הטריגר לפונקציה לטובת ההרצה:



ונצפה בלוגים שנוצרו:



נוכל לראות בתמונה מעל את הכתובת IP שממנה יצאנו בעת ביצוע ההתקפה שהינה משוייכת ל-Microsoft וכמובן גם המיקום שונה, נוכל להשתמש בטכניקה זו לטובת Password Spray, Web Crawling, C&C וכו'.

כמו במקרה הקודם זה יכול לעזור לנו לא להיחסם מכיוון שהכתובת IP הינה לגיטימית ובנוסף אנו לא נחשוף את הכתובת IP שלנו ואת המיקום, חשוב לציין שזה יכול לעזור לנו בעקיפת מנגנון Smart Lockout שבין היתר בודק האם הכתובת IP שמגיעות ממנה בקשות מסומנת כזדונית.

הגנות

אז מה זה Microsoft Entra ID Protection?

Microsoft Entra ID Protection הינו שירות אבטחה מתקדם המסייע לארגונים לזהות, לחקור ולצמצם סיכונים מבוססי זהות באמצעות יכולות למידה מתקדמות.

השירות מנתח את התנהגות המשתמשים ודפוסי התחברות על סמך אותו שונים כגון כתובות IP, מיקומים גאוגרפים, מכשירים וכו', במטרה לזהות איומים פוטנציאלים.

מהו סיכון? (Risk)

סיכון מתייחס להערכת פעולות משתמשים, תהליכי אימות והמאפיינים הנלווים אליהם.

Microsoft Entra ID Protection מזהה סיכונים מבוססי זהות ומספק דוחות ייעודיים, המאפשרים לצוותי אבטחה לזהות במהירות איומים ולהגיב אליהם באופן אפקטיבי.

מהם דוחות הסיכון של ID Protection?

ID Protection מפיק דוחות על משתמשים בסיכון, התחברויות בסיכון, Workload Identities בסיכון ו- Risk Detections בסיכון בארגון. חקירת דוחות אלו תסייע בחיזוק מערך האבטחה באמצעות זיהוי וטיפול בחולשות במנגנוני האימות ובקרות הגישה.

Smart Lockout

מה זה Smart Lockout?

Smart Lockout הינו מנגנון אבטחה שנועד להגן על חשבונות משתמשים מפני גישה בלתי מורשית, באמצעות חסימת תוקפים המנסים לבצע מתקפת Password Spray או Brute Force.

המנגנון יודע להבחין בין ניסיונות התחברות לגיטמיים של משתמשים לבין ניסיות חשודים ממקורות לא מוכרים.

בעוד שהתוקפים נחסמים לאחר מספר ניסיונות כושלים, משתמשים לגיטמיים יכולים להמשיך ולהתחבר ללא פגיעה בזמינות.

כיצד Smart Lockout פועל?

- 10 ניסיונות כושלים ל-Azure Public או Microsoft Azure.
- 3 ניסיונות כושלים עבור Azure US Government.

לאחר החסימה הראשונית, כל ניסיון כושל נוסיף מאריך את משך החסימה, החל מדקה אחת ובהתארכות הדרגתית לאורך זמן.

כדי למנוע חסימות מיותרות, Smart Lockout עוקב אחר שלושת ניסיונות ההתחברות השגויים האחרונים, הזנה חוזרת של אותה סיסמה שגויה לא תגדיל את מונה החסימות.

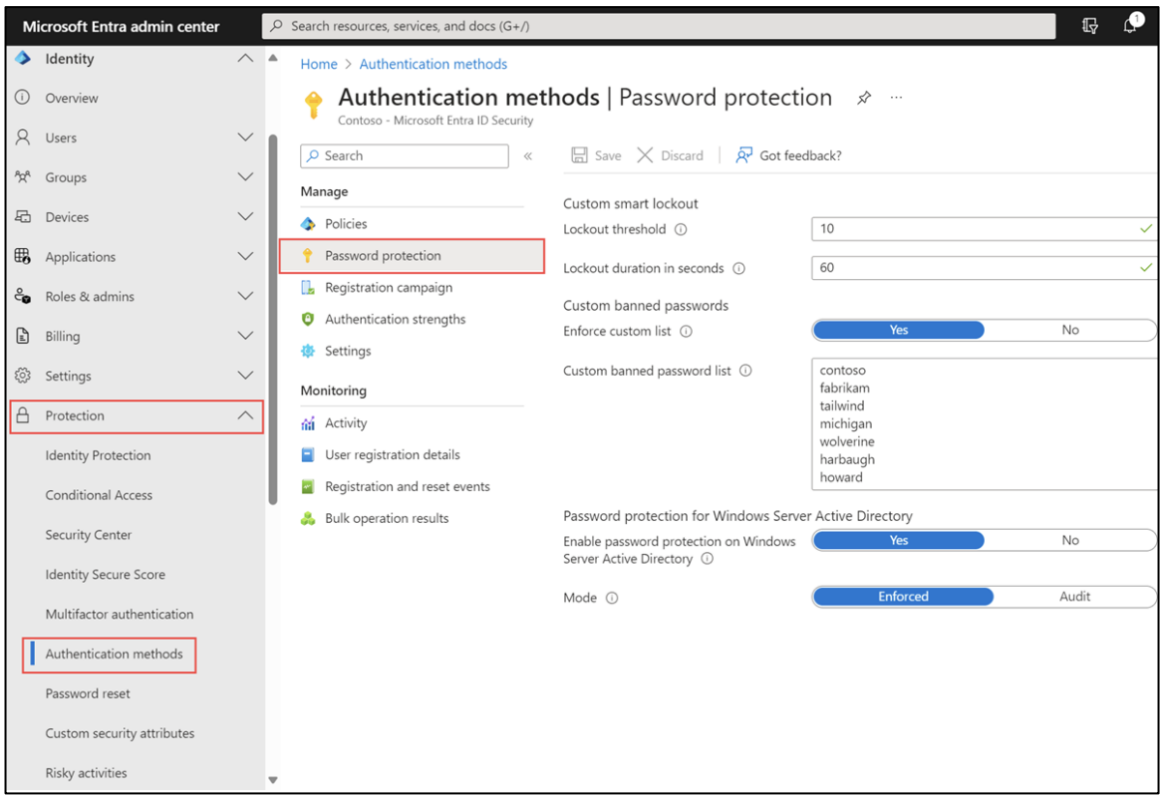
מה קורה כאשר Smart Lockout מופעל?

כאשר מגיעים לסף החסימה של Smart Lockout, המשתמשים יקבלו את ההודעה הבאה:

"Your account is temporarily locked to prevent unauthorized use. Try again later, and if you still have trouble, contact your admin."

מאחר ותהליכי האימות של Microsoft Entra מבוזרים גאוגרפית ומנוהלים באמצעות איזון עומסים (Load Balancing) נסיונות התחברות עשויים להיות מעובדים במרכזי נתונים שונים.

התמונה הבאה מתארת את ה-Password Protection Settings:



יכולת זיהוי מתקפות Password Spray בזמן אמת, זמינה החל מינואר 2025 תחת Microsoft Entra ID Protection, מה שמאפשר זיהוי ומניעה ביטווח המיידי של מתקפות Password Spray כבר במהלך תהליך ההתחברות ובכך מונע מתוקפים קבלת Tokens ומקצר את זמן התגובה והטיפול מאירוע שנמשך שעות לשניות בודדות.

על מנת להשתמש ביכולת זו, יש לוודא כי Microsoft Entra ID Protection מופעל בארגון.

ניתן להגדיר מדיניות Conditional Access מבוססת סיכון (Risk-based Conditional Access) שתפעל באופן אוטומטי עם זיהוי ניסיונות Password Spray באמצעות העלאת רמת הסיכון של הסשן והפעלת אפשרויות אימות נוספים או חסימת ניסיונות התחברות שהינם חשודים.

למידע נוסף על הגדרה, יישום וניהול של יכולת זו:

What is Microsoft Entra ID Protection? - Microsoft Entra ID Protection | Microsoft Learn

סיכום

במאמר זה עברנו על האפשרויות להשתמש בפונקצינליות של משאבים לגיטימים ב-Azure על מנת לבצע תרחיש תקיפה שיצא מכתובת IP שהינה לגיטימית, ממיקום שונה ממה שאנו נמצאים בו ויעזור לנו להתגבר בין היתר על מנגנוני הגנה אשר מוודאים את המקור שאינו זדוני.

על המחבר

אני מתן בכר, חי את עולם התשתית והענן בשנים האחרונות ופשוט נהנה ממה שאני עושה.

מוזמנים לפנות אלי בלינקדאין או ב-Twitter.

מקורות מידע

- What is Microsoft Entra ID Protection? - Microsoft Entra ID Protection | Microsoft Learn
- Download Azure IP Ranges and Service Tags – Public Cloud from Official Microsoft Download Center