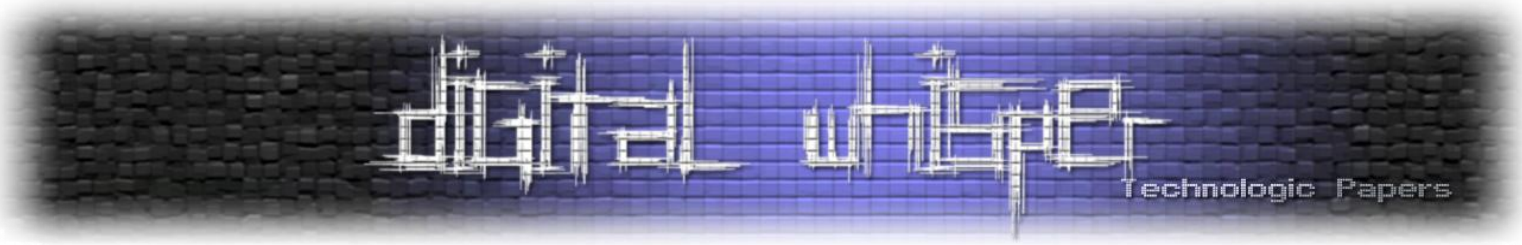


Permission Impossible: Azure API Vulnerabilities and Over-Privileged Roles

מאת אריאל סימון

הקדמה

האם אתם יכולים לבטוח בספקית הענן שלכם? 🤔

במאמר הזה אציג את המחקר שעשיתי על Azure, שגילה מספר בעיות אבטחה:

- מספר Role-ים שהוגדרו לא נכון ע"י המפתחים של Azure - ונותנים יותר הרשאות ממה שהם אמורים לתת
- חולשת API המאפשרת לתוקף להזליג מפתחות ל-VPN הארגוני
- איך שילוב של הבעיות האלו יוצר מתאר תקיפה חדש, שמאפשר **למשתמש חלש** בסביבת הענן להשיג גישה לרשתות ארגוניות - למשאבים בענן, ל-VPC-ים, ואפילו לרשתות On-prem

הממצאים של המחקר הזה, ובעיקר התגובה של Azure לדיווח על הבעיות, מעלים שאלות של איך וכמה אני יכול לבטוח בשירותים הדיפולטים שניתנים לי.

ועל הדרך גם נלמד על מודלי הרשאות, שיטות תקיפה בענן, ועוד שטיקים של Azure.

אז יאללה בואו נתחיל.

מה זה Azure RBAC?

עבור כל מי שלא התעסק עם Azure בעבר (מה אתם לא אוהבים חולשות?!), נעשה סקירה קצרה של המושגים הבסיסיים.

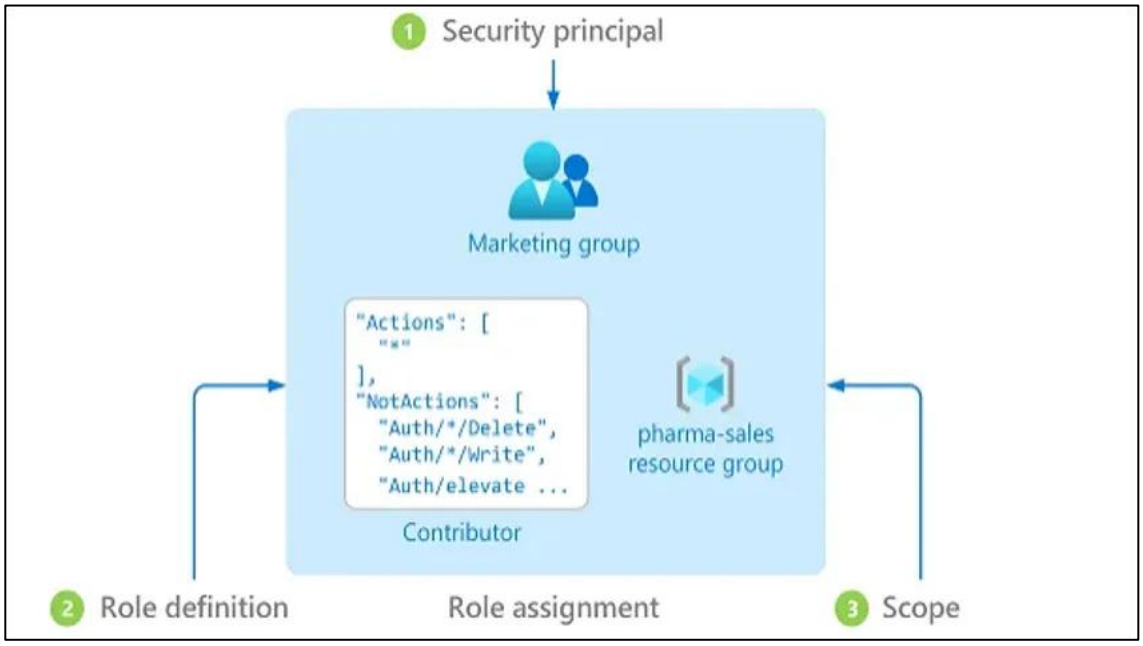
מודל ההרשאות בAzure, Azure RBAC (Role-based Access Control), כשמו כן הוא, מבוסס על Role-ים.

Role הוא בעצם קבוצת הרשאות שאפשר לתת ל-Principal (Principal יכול להיות user, group, service principal, managed identity וכו').

כשמעניקים Role ל-Principal, נוצר Role Assignment, שזה אובייקט שמכיל שלושה מרכיבים מרכזיים:

1. **Principal** - מי מקבל את ההרשאות?
2. **Role definition** - איזה Role ניתן? מה ההרשאות שהוא מעניק?
3. **Scope** - לאיזה משאבים ה-Role assignment הזה מעניק גישה? יש כמה אופציות של רמות ב-Scope. ה-Scope יכול להיות רחב, כמו Management group שלם (שמכיל כמה subscriptions), או יותר ספציפי, כמו Resouce group, או Resource ספציפי (כמו מכונה וירטואלית, דאטאבייס וכו').

* לא הסברתי על מושגים בסיסיים כמו Management group ו-Subcription, אם תרצו לקרוא עליהם מוזמנים להיכנס לכאן.



Azure roles

בAzure, יש יותר מ-400 Built-in roles, אשר נוצרו ע"י המערכת ומוכנים לשימוש דיפולטית. אפשר לחלק אותם לשתי קטגוריות:

1. **Generic** - Role-ים שנותנים הרשאות לכל המשאבים והשירותים של Azure ב-Scope (למשל Contributor, Owner, Reader וכו')

2. **Service-specific** - Role-ים שנותנים הרשאות לשירות ספציפי או משאב מסוג מסוים ב-Scope (למשל Virtual Machine Contributor)

לדוגמה, אם אתם מעניקים את ה-Role שנקרא Contributor על Scope של Subscription, אתם נותנים את ההרשאות האלו לכל המשאבים ב-Subscription הזה, ללא תלות באם יש שם VM-ים, Storage accounts, Databases, או כל משאב אחר. אבל אם אתם נותנים את Virtual Machine Contributor, אתם בעצם תתנו הרשאות רק ל-Virtual Machines שבאותו Subscription.

ניתן לראות את ה-Tradeoff כאן - להשתמש ב-Service specific roles היא הגישה היותר מאובטחת, בגלל שאתם נותנים הרשאות בצורה יותר ממוקדת, ופחות הרשאות כלליות. אבל מן הסתם שיותר קל להשתמש בGeneric roles, כי בגישה הזאת צריך להתעסק בפחות מתן וניהול הרשאות.

הבעיה

בואו נסתכל על כמה Role-ים ועל ההרשאות שהם נותנים. נסו לראות אם אתם מצליחים לקלוט איפה דברים משתבשים.

Reader role

אחד ה-Built-in Roles בAzure שהם מהסוג הגנרי, נקרא Reader. כמו שהייתם מצפים, הוא נותן הרשאות קריאה בלבד לכל המשאבים ב-Scope הנבחר. בואו נסתכל על ההגדרה שלו (Role definition JSON):

```
{
 "assignableScopes": [
  "/"
 ],
 "description": "View all resources, but does not allow you to make any changes.",
 "id": "/providers/Microsoft.Authorization/roleDefinitions/acdd72a7-3385-48ef-bd42-f606fba81ae7",
 "name": "acdd72a7-3385-48ef-bd42-f606fba81ae7",
 "permissions": [
  {
   "actions": [
    "*/read"
   ],
   "notActions": [],
   "dataActions": [],
   "notDataActions": []
  }
 ],
 "roleName": "Reader",
 "roleType": "BuiltInRole",
 "type": "Microsoft.Authorization/roleDefinitions"
}
```

כמו שאנחנו רואים בשדה ה-actions, ההרשאה שניתנת היא **/read***, שתכל'ס אומרת שאנחנו יכולים לקרוא כל דבר ב-Scope.

* שימו לב שלא מדובר ב-Data actions (קריאה של אובייקטי דאטא, כמו למשל קבצים ב- Storage accounts, Secret-ים ב-Key vault וכו'). אלו דורשים הרשאות מיוחדות אחרות, שהן יותר רגישות.

אוקיי, אז יש לנו Role גנרי שנותן הרשאות קריאה גנריות, הגיוני.

מה לגבי הרשאות של Service-specific roles?

Workbook Reader

```
{
 "assignableScopes": [
  "/"
 ],
 "description": "Can read workbooks.",
 "id": "/providers/Microsoft.Authorization/roleDefinitions/b279062a-9be3-42a0-92ae-8b3cf002ec4d",
 "name": "b279062a-9be3-42a0-92ae-8b3cf002ec4d",
 "permissions": [
  {
   "actions": [
    "microsoft.insights/workbooks/read",
    "microsoft.insights/workbooks/revisions/read",
    "microsoft.insights/workbooktemplates/read"
   ],
   "notActions": [],
   "dataActions": [],
   "notDataActions": []
  }
 ],
 "roleName": "Workbook Reader",
 "roleType": "BuiltInRole",
 "type": "Microsoft.Authorization/roleDefinitions"
}
```

אם נסתכל שוב על ה-actions, נראה שכמו שאומר התיאור, ה-Role נותן הרשאות לכמה פעולות שקשורות ל-Workbooks.

אז Service-specific role, שנותן הרשאות ל-Service ספציפי! בינתיים הכל נורמלי.

Managed Applications Reader

עכשיו, בואו נסתכל על Managed Applications Reader, שהתיאור שלו הוא:

"Lets you read resources in a managed app and request JIT access."

```
{
 "assignableScopes": [
  "/"
 ],
 "description": "Lets you read resources in a managed app and request JIT access.",
 "id": "/providers/Microsoft.Authorization/roleDefinitions/b9331d33-8a36-4f8c-b097-4f54124fdb44",
 "name": "b9331d33-8a36-4f8c-b097-4f54124fdb44",
 "permissions": [
  {
   "actions": [
    "Microsoft.Resources/deployments/*",
    "Microsoft.Solutions/jitRequests/*",
    "*/read"
   ],
   "notActions": [],
   "dataActions": [],
   "notDataActions": []
  }
 ],
 "roleName": "Managed Applications Reader",
 "roleType": "BuiltInRole",
 "type": "Microsoft.Authorization/roleDefinitions"
}
```

אז כהרגלנו, נסתכל על ה-actions - ניתן לראות שה-Role נותן גישה ל-Deployments, ל-JIT Requests, ו...

לקרוא הכל???

אז ה-Role הזה, שאמור לתת גישה לקרוא Managed Applications ול-JIT, בתכל'ס נותן גם גישה לקרוא כל משאב בAzure?

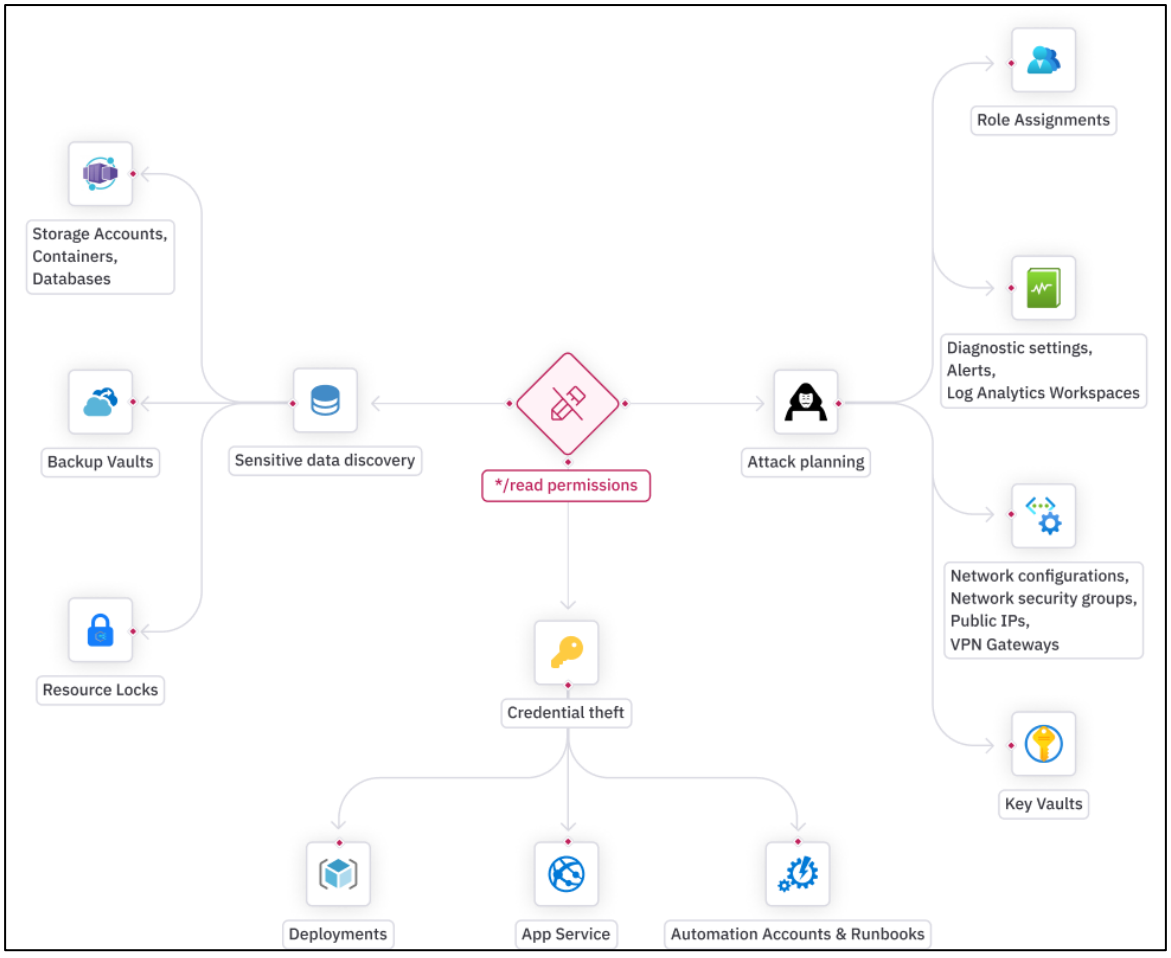
אבל זה לא מה שהתיאור אומר.. ובטח לא מה שמישהו שמשתמש ב-Role הזה היה מצפה!

בשורה התחתונה, השם והתיאור של ה-Role הזה מטעים את המשתמש, וגורמים לו לחשוב שהוא נותן הרשאות ספציפיות, כשלמעשה **הוא נותן הרשאות גנריות לכל המשאבים.** תוקף שמשיג גישה למשתמש עם ההרשאות האלו, יכול לעשות הרבה דברים. בואו נבין מה בדיוק.

כמה זה גרוע?

כשראיתי את זה חשבתי לעצמי, טוב.. זה רק הרשאות קריאה, כמה גרוע זה יכול להיות? טעיתי.

בואו נצלול למה באמת אפשר לעשות עם ההרשאה הזאת, וכמה שימושי זה יכול להיות לתוקף. בגלל שיש כל כך הרבה פעולות, חילקתי אותן ל-3 קטגוריות ונתתי קצת דוגמאות לכל אחת:



Credential Theft:

- **Automation Accounts, Deployment scripts, Web applications** - האחד הזה ממש הפתיע אותי. ההרשאה הזאת מאפשרת לנו לקרוא קוד מקור ומשתני סביבה של סקריפטים ואפליקציות. המשותף לשלושת ה-service-ים שציינתי כאן, הוא שהם כולם עושים פעולות מול סביבת ה-Azure, מה שהופך את הסיכוי שיהיה בהם Credentials לגבוה מאוד!

Sensitive data discovery:

- **Storage accounts, container registries, databases** - לזהות את כל המשאבים ואת ה-Metadata שלהם כדי למצוא איפה מאוחסן המידע הרגיש שהתוקף מחפש
- **Resource locks** - פיצ'ר Azure-י שנותן "לנעול" משאבים כדי למנוע מחיקה או שינוי שלהם. לתוקף זה שימושי כדי לזהות אילו משאבים הם קריטיים ורגישים (אם מישהו טרח לנעול Database אחד מתוך עשרים, זה בטח האחד החשוב)
- **Backup vaults** - למצוא גיבויים של DB-ים, של Storage accounts וכו'

Attack planning:

- **Role assignments** - לדעת מי יכול לגשת לאילו משאבים. שימושי כדי למפות ולתכנן נתיבי תקיפה ו-Privilege escalation
- **Diagnostics settings, alerts, log analytics workspaces** - לדעת איזה לוגים נשמרים ולאן, ואיזה התראות הוגדרו. שימושי לתוקף כדי להמנע מהתגלות
- **Network configurations, network security groups, VPN gateways** - כל הגדרות הרשת המאפשרות לתכנן את נתיבי הפעולה, לדעת איזה חיבורים ניתן לעשות ואילו לא
- **Key vaults** - לזהות את כל ה-Vaults ואת הפרטים שלהן

כמה רחב זה?

אז אתם בטח חושבים לעצמכם: "אוקיי זה מגניב והכל... אבל אם אני לא משתמש ב Managed Applications Reader role... אז זה לא משפיע עליי, נכון?"
אז תחשבו שוב.

אחרי שעברתי על כל ה-Built-in roles בAzure, מצאתי שהבעיה הזו (ההמצאות של הרשאת */read בלי שום סיבה הגיונית) חוזרת ב-10 Role-ים שונים (!!!), ראו בטבלה בעמוד הבא את הפירוט.

שם ה-Role	הרשאות שה-Role נותן
Log Analytics Reader	***/read** Microsoft.OperationalInsights/workspaces/analytics/query/action Microsoft.OperationalInsights/workspaces/search/action Microsoft.Support/*
Log Analytics Contributor	***/read** Microsoft.ClassicCompute/virtualMachines/extensions/* Microsoft.ClassicStorage/storageAccounts/listKeys/action [...]
App Compliance Automation Administrator	***/read** Microsoft.AppComplianceAutomation/* Microsoft.Storage/storageAccounts/blobServices/write [...]
App Compliance Automation Reader	***/read**
Managed Application Contributor Role	***/read** Microsoft.Solutions/applications/* [...]
Managed Application Operator Role	***/read** Microsoft.Solutions/applications/read Microsoft.Solutions/*/action
Managed Applications Reader	***/read** Microsoft.Resources/deployments/* Microsoft.Solutions/jitRequests/*

Monitoring Contributor	***/read** Microsoft.AlertsManagement/alerts/* Microsoft.AlertsManagement/alertsSummary/* [...]
Monitoring Reader	***/read** Microsoft.OperationalInsights/workspaces/search/action Microsoft.Support/* [...]
Resource Policy Contributor	***/read** Microsoft.Authorization/policyassignments/* Microsoft.Authorization/policydefinitions/* [...]

הסיכון הזה קיים בכל User, Service Principal, Managed Identity או חבר ב-Group שיש לו אחד מה-Role-ים ה"תמימים" האלו.

כמו שניתן לראות בטבלה, לחלק מה-Role-ים האלה יש הרשאות קריאה יותר ספציפיות בנוסף ל-*/read, כמו למשל הרשאת ה- Microsoft.Solutions/applications/read ב- Managed Application Operator Role, שבעצם כבר כלולה ב-*/read. זה מראה שאחת ההרשאות האלה היא מיותרת, ושהמפתח שהוסיף אותה עשה את זה בטעות, או מתוך עצלות ("אה זה לא עובד? טוב תוסיף שם כוכבית זה בטח יעבוד..").

המקרה של App Compliance Automation Reader הוא אפילו יותר הזוי, כי יש לו **רק** את הרשאת ה-*/read, מה שהופך אותו ל**זהה לחלוטין ל-Role הגנרי, החזק והאימתני שנקרא Reader!**

אז זה די רע... אבל אולי אנחנו יכולים לעשות את זה רע אפילו יותר?

חולשת VPN

רציתי להפוך את מתאר התקיפה הזה לאפילו יותר חזק, ולמצוא עוד דברים מוזרים וחזקים שאפשר לעשות עם */read. אז הסתכלתי על התיעוד של ההרשאה:

Actions	Description
*/read	Read resources of all types, except secrets.

אז נראה שמשתמשים בעלי הרשאות קריאה לא אמורים להיות מסוגלים לקרוא Secret-ים (ואם נחשוב על זה רגע, זה די הגיוני, כי ה-Secret-ים האלו יכולים לשמש כדי להשיג הרשאות נוספות מעבר לקריאה, לגשת לעוד משאבים וכו').

רציתי להבין מה המשמעות של הכוכבית בהרשאה שלנו. ברור שכוכבית אומר 'הכל', אבל אילו פעולות מסתתרות שם? כמה פעולות כאלה יש?

השתמשתי בכלי נחמד בשם AzAdvertizer כדי לחפש בצורה נוחה על כל ההרשאות בAzure, וחיפשתי כל הרשאה שנגמרת ב-'/read'. גיליתי שיש לא פחות מ-**9618** פעולות שכלולות בביטוי */read! וואו.

אז המחשבה שלי הייתה:

אם אני אוכל למצוא פעולה אחת, מתוך ה-9618, שתאפשר לי להדליף Secret, תהיה לי פה חולשה רצינית!

אחרי שעברתי כל הפעולות (בסדר בסדר, יכול להיות שהשתמשתי ב-ChatGPT), מצאתי פעולה אחת שנראתה לי מעניינת:

Gets Vpn Link Connection Shared Key	microsoft.network/vpnGateways/vpnConnections/vpnLinkConnections/sharedKeys/read

מה זה VPN Link? מה זה Connection? מה זה Shared key? אין לי מושג, אבל יש בזה את המילה key, אז זה חייב להיות מעניין :)

הבאג

אז יש לנו קריאת API שמחזירה Secret כלשהו למרות שיש לי רק הרשאות קריאה. אבל למה זה קורה? מה הטעות שהמפתח עשה כאן?

ב-Azure, קריאות API ממומשות בעזרת מתודות HTTP שונות. לדוגמה, הקריאה Virtual Machines - List ממומשת עם GET, והקריאה Virtual Machines - Install Patches ממומשת עם POST. זה הגיוני, כי ה-API של Install Patches צריך לקבל מידע מהמשתמש, כמו איזה Patch להתקין (כלומר צריך גוף לבקשה ולכן זו בקשת POST), בעוד שה-API של List VMs רק מושך מידע מהשרת, ולא צריך להעביר מידע מהמשתמש (ולכן זו בקשת GET).

אבל מה שמצאתי בעזרת קצת מחקר Blackbox, קריאת דוקומנטציה, והרבה ניסוי וטעייה, זה ש-Azure בחרו לממש את אכיפת ההרשאות בעזרת שינוי של מתודת ה-HTTP של בקשות ה-API.

נשמע מוזר לא? אני אסביר:

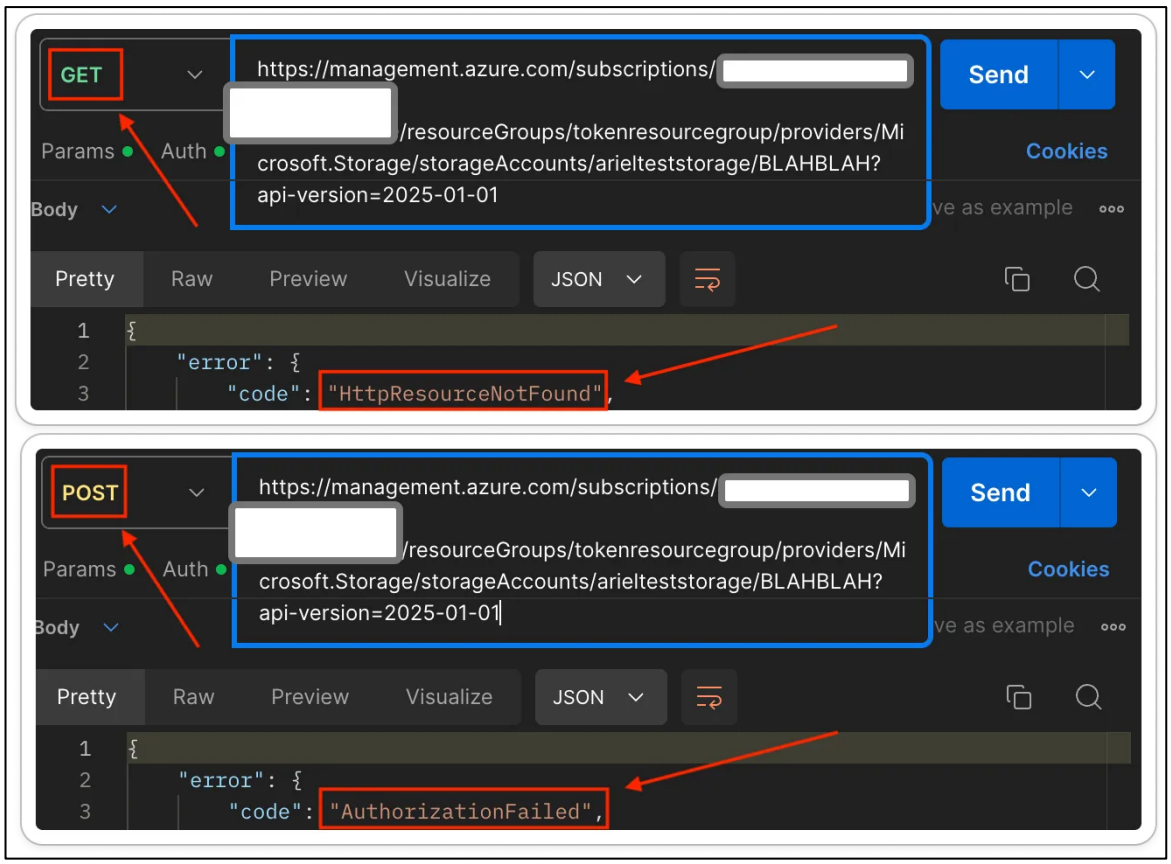
נראה שמשתמשים עם הרשאות קריאה בלבד (כמו */read) יכולים לשלוח בקשות GET לשרת, אבל יקבלו Access denied אם ינסו לשלוח בקשות POST.

פעולות קריאה רגילות, כמו Virtual Machines - List ממומשות עם GET כמו שראינו, אבל פעולות קריאה של ערכים רגישים ו-Secret-ים, כמו Storage Accounts - List Keys או Database Accounts - List Connection Strings ממומשות עם POST. ולמה זה מוזר? כי אין בהם גוף לבקשה!

כמו שאתם בטח מבינים, זה ממומש בצורה הזו כדי לוודא שאכיפת ההרשאות אכן פועלת, ושמשתמשים בעלי הרשאות קריאה בלבד לא יוכלו לגשת ל-API-ים רגישים.

כדי להוכיח את זה, ביצעתי קריאת API ל-Endpoint שלא קיים, עם משתמש בעל הרשאות קריאה בלבד. כשאני שולח את הבקשה עם מתודת GET, אני מקבל את השגיאה HttpResourceNotFound . אבל כשאני שולח את אותה הבקשה לאותו ה-Endpoint הלא קיים, הפעם עם POST, אני מקבל את השגיאה AuthorizationFailed . זה מוכיח שבדיקת ההרשאות נעשית בעזרת הסתכלות על המתודה, ולא בעזרת בדיקה של ה-API Endpoint הספציפי שפניתי אליו, והאם באמת יש לי הרשאות אליו.

וככה זה נראה:

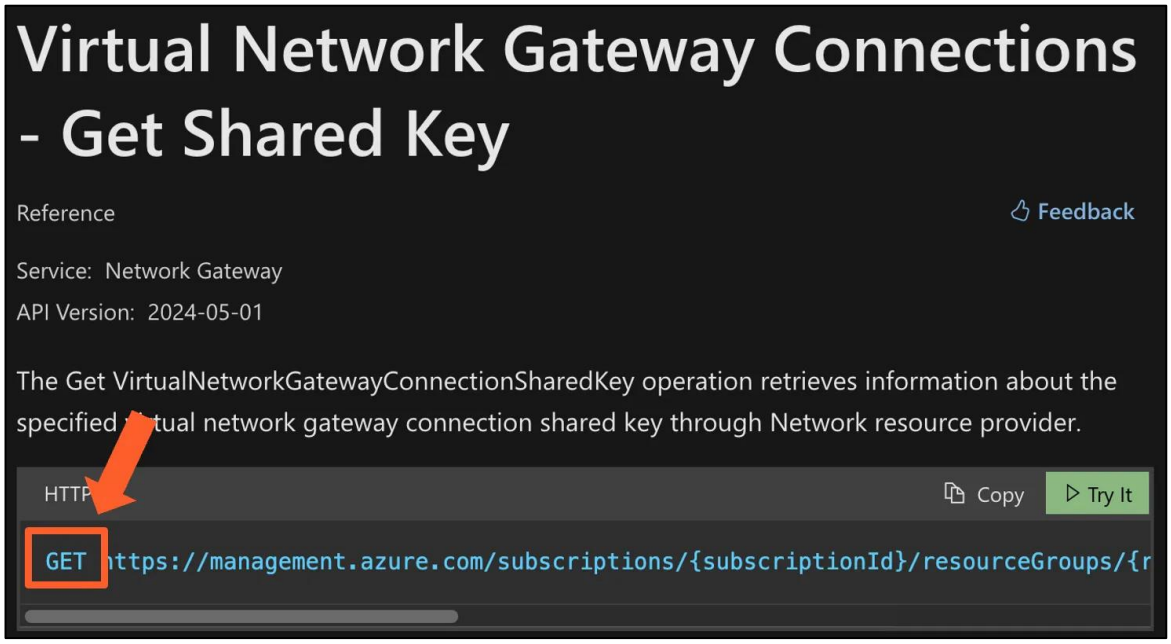


אז למה זה כזה בעייתי אתם שואלים?

כי כשאתם בוחרים לממש את המערכת שלכם בדרך שהיא לא הגיונית ולא אינטואיטיבית, גם המפתחים שלכם יכולים להתבלבל ולעשות טעויות קריטיות...

ההנחה שלי היא שמתישהו, מפתח כלשהו ב-Azure התבלבל ומימש קריאת API חדשה שמחזירה Secret, עם המתודה שהכי הגיונית לשימוש בסיטואציה הזאת, שהיא כמובן GET, כי אין גוף לבקשה. וכשהוא עשה את זה, בלי לשים לב הוא יצר חולשה.

אם נסתכל על ה-API שמצאנו קודם, נראה שהתאוריה שלנו הייתה נכונה! הוא מומש בטעות עם GET, מה שמאפשר למשתמשים בעלי הרשאות read-only לשלוף את המפתח!



אז עכשיו יש לנו חולשה. כל מה שנשאר זה להבין כבר מה זה המפתח הזה!

Azure VPN Gateway

VPN Gateway הוא שירות Azure-י שממממש VPN ומאפשר למשתמשים לחבר את הרשתות המרוחקות שלהם זו לזו. ארגונים משתמשים בשירות הזה כדי לתמוך סביבות היברידיות (לחבר בין רשת ה-On-prem ל-Cloud), וגם כדי לחבר בין סביבות On-prem (נקרא גם Site-to-site, או S2S). משתמשי קצה גם יכולים להתחבר לרשת, כדי לאפשר סיטואציות כמו עבודה מרחוק (נקרא גם Point-to-site, או P2S).

חיבורי P2S דורשים התאמתות חזקה, בעזרת Certificate, Radius server, או התאמתות Entra ID. אבל חיבורי Site-to-site לעומת זאת, דורשים אך ורק את ה-**Pre shared key (PSK)** שהוא סיסמא שמשותפת בין ה-VPN Device של כל Site (למשל ה-Fortigate הארגוני), ובין השירות של Azure VPN Gateway.
וכן, זו הסיסמא שאנחנו יכולים לשלוף עם החולשה שלנו!

מתאר התקיפה המלא

1. התוקף משיג גישה לזהות חלשה (בעלת הרשאות קריאה בלבד, או אחת שיש לה את אחד מה- Over-privileged roles שמצאנו קודם).
2. התוקף שולף את ה-Pre shared key של שירות VPN Gateway.
3. בעזרת המפתח, התוקף מתחבר ל-S2S Connection, ואז הוא נגיש לרשת הפנימית, כולל VPC-ים ורשתות מקומיות (on-prem) של הארגון הנתקף.
   - הדגמה: https://www.youtube.com/watch?v=eKhZKH0p8BI

הסרטון מציג איך משתמש ללא הרשאות כלל, שמעניקים לו את ה-Log Analytics Reader role (שאמור לתת גישה רק לקריאת לוגים, אבל כמו שאנחנו כבר יודעים, הוא הרבה יותר מזה) יכול פתאום לשלוף את ה-VPN PSK, ואז משתמש בו כדי להתחבר לרשת הארגונית.

עם הגישה הזו, התוקף יכול ליצור Site זדוני, ולהתחבר למשאבי ענן ו-Site-ים אחרים, ולרשתות On-prem.

כתלות בקונפיגורציה, במקרים מסוימים זה יכול לעבוד כשהחיבור מגיע מה-IP שמקונפג ב-VPN Gateway. במקרה הזה, תוקף שהשיג גישה לרשתות ה-On-prem של ארגון יכול להשתמש במתאר הזה כדי להשיג גישה למשאבי הענן, ל-VPC-ים, ול-Site-ים אחרים.

אגב, יש אפילו עוד ערכים רגישים שאפשר לגשת אליהם בעזרת הרשאות Read בלבד. מחקר מצוין של Binary Security מראה איך בעזרת הרשאות Read אפשר לבצע Privilege escalation בשירות Azure API Management ע"י שליפה של מפתחות, מה שמוביל להשתלטות מלאה על השירות.

מה יש ל-Azure לומר?

דיווחתי למייקרוסופט על שתי הבעיות:

Over-privileged roles

מייקרוסופט קבעו שהבעיה הזאת היא בדירוג 'low severity' והם החליטו לא לתקן אותה.
אחרי כמה ימים ראיתי שינויים נרחבים בתיעוד שמבוססים על הדיווח שלי: התיעוד של כל עשרת ה-Role-ים שדיווחתי עליהם השתנה והתווסף אליהם המשפט הבא:

Log Analytics Reader

Log Analytics Reader can view and search all monitoring data as well as and view monitoring settings, including viewing the configuration of Azure diagnostics on all Azure resources.

This role includes the `*/read` action for the control plane. Users that are assigned this role can read control plane information for all Azure resources.

אז הם בחרו לתקן את התיעוד במקום לתקן את הבעיה, שעדיין מסכנת את המשתמשים.

חולשת VPN

כאן התגובה הייתה שונה, מייקרוסופט הכירו בבעיה כחולשה בדירוג 'Important severity' והם תיקנו אותה. בנוסף הם נתנו לי Bounty של 7500$.

הייתי שמח לגלות שהתיקון של החולשה לא היה לשנות את ה-HTTP Method ל-POST. בדף התיעוד החדש שמסביר את ההרשאות של שירות ה-VPN, נאמר שבשביל לשלוף את ה-PSK, דרושה ההרשאה Microsoft.Network/connections/sharedKey/action החדשה:

Connection/Preshared Key	Update existing	Microsoft.Network/connections/sharedKey/action

זה תיקון מעולה!

טיפים למגנים

אל תשתמשו ב-Role-ים הבעייתיים

כמו שראינו, בעיית ה-Over-privileged roles לא הולכת להיות מתוקנת. המנעו משימוש ב-Role-ים האלו בסביבה שלכם, ותשתמשו באלטרנטיבות (שנציג מיד).

תקפידו על Scope-ים מצומצמים

במקום ליצור Role assignments עם Scope רחב, כמו Management group או Subscription, תגבילו אותם למשאב או ל-Resource group ספציפיים - רק מה שהזהות באמת צריכה גישה אליו!

תבנו Custom roles

במקום להשתמש ב-Built-in Roles שלא מנוהלים על ידיכם, תשתמשו ב-Custom roles שאתם בונים, ותתנו בהם רק את ההרשאות שבאמת נדרשות. תחליפו את ה-Role assignments הנוכחיים שלכם (לפחות את אלה שמשתמשים ב-Role-ים הבעייתיים) ב-Custom roles עם הרשאות נכונות.

סיכום

לאבטח סביבות ענן זה קשה.

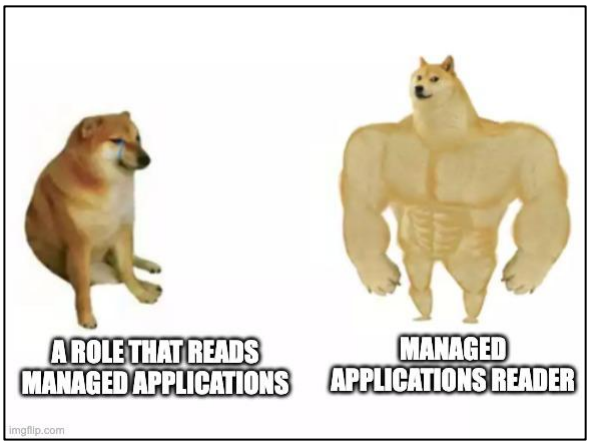
ה-Shared responsibility model ("מודל האחריות המשותף") שכל ספקיות הענן מציגות ותומכות בו, מחלק את האחריות האבטחתית בין ספקית הענן ובין המשתמשים, ומגדיר איזה משימות אבטחה הן באחריות המשתמש, ואיזה משימות אבטחה הן באחריות ספקית הענן.
אבל הבעיות שדיברנו עליהן כאן הן נופלות בין הכיסאות: כשספקית הענן נותנת למשתמשים שירות שאמור לעזור להם בניהול זהויות והרשאות, אבל בפועל מטעה אותם וגורם להם לעשות טעויות מסוכנות, מי האשם? האם זאת ספקית הענן שגרמה למשתמש ליצור את הבעיה, או שזה המשתמש, שבאמת יצר אותה מבלי לבדוק מה הוא עושה?

אין תשובה קלה פה, אבל דבר אחד בטוח - אסור לבטוח באופן עיוור בשירותים שנותנים לנו. תמיד צריך לבדוק, לקרוא תיעוד, ולוודא שדברים עובדים כמו שאנחנו מצפים.

על המחבר

אני אריאל סימון, חוקר אבטחה ב-Token Security, מתעסק במחקר חולשות ב-Cloud, מרצה בכנסים ואוהב מסיבות 🎉

מוזמנים לפנות אליי בכל נושא ב-LinkedIn.



מקורות מידע

- https://learn.microsoft.com/en-us/azure/role-based-access-control/overview
- https://learn.microsoft.com/en-us/azure/role-based-access-control/scope-overview
- https://learn.microsoft.com/en-us/azure/role-based-access-control/built-in-roles
- https://learn.microsoft.com/en-us/azure/vpn-gateway/roles-permissions
- https://www.azadvertizer.net/
- https://binarysecurity.no/posts/2024/11/apim-privesc