

Group Policy 101

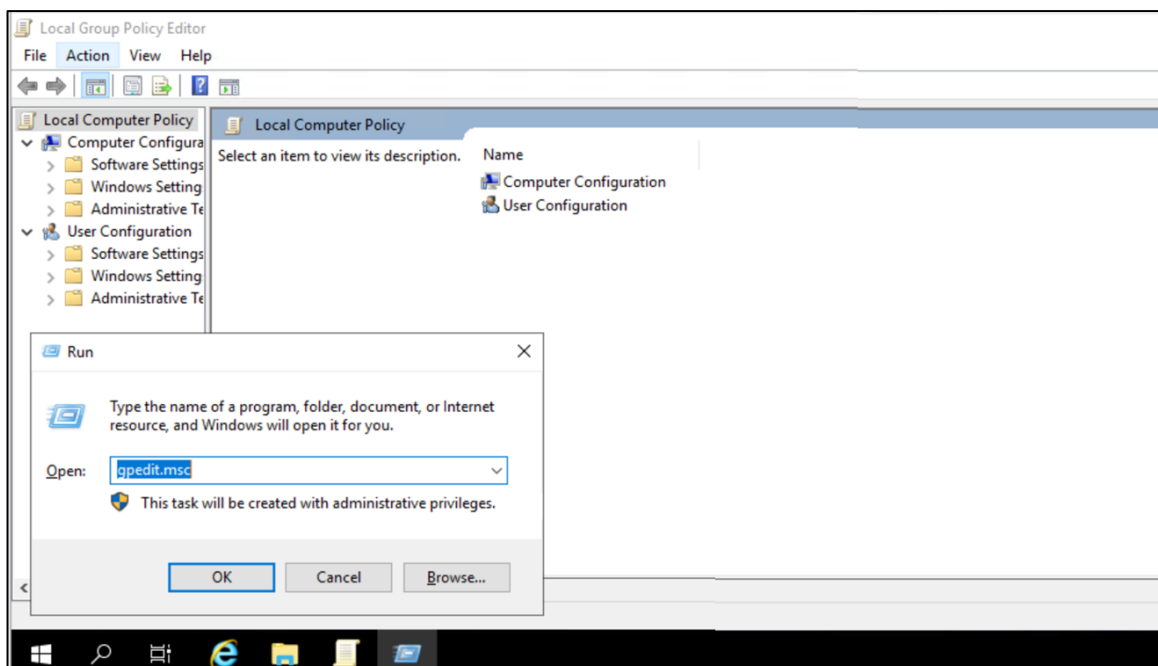
מאת ספיר פדרובסקי

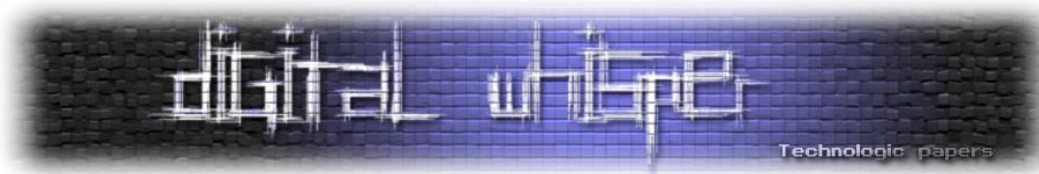
הקדמה

אם התעסקתם אי פעם בתחום ההגנתי בעולמות ה-Windows סביר ששמעתם על GPO. אבל גם אם לא, אל חשש! מדובר בנושא פשוט להבנה, אך עד כמה שהוא פשוט, כך הוא גם מסוכן. במאמר זה, אסביר על GPO, מה הם, מה ניתן לעשות איתם (באופן לגיטימי), מה ניתן לעשות איתם (באופן פחות לגיטימי), איך נוכל לזהות שימוש זדוני ב-GPO ועוד כמה דברים כיפיים.

אז נתחיל מהבסיס, מה זה GPO?

בתרגום ישיר, Group Policy (מדיניות קבוצה). בגדול - מדובר בסט של חוקים שניתן להחיל על משתמשים, קבוצות ומכונות. כל-GPO (Group policy object) מיוצג על ידי-GUID, חד ערכי. ניתן גם לתת שם ל-GPO אך זה לא חובה. ה-GPO יכול סט חוקים או הגדרות למערכת הקבצים ו/או ל-Active Directory. כדי להגדיר GPO, באמצעות ה-Local Group Policy Editor. הריצו את הפקודה: gpedit.msc:

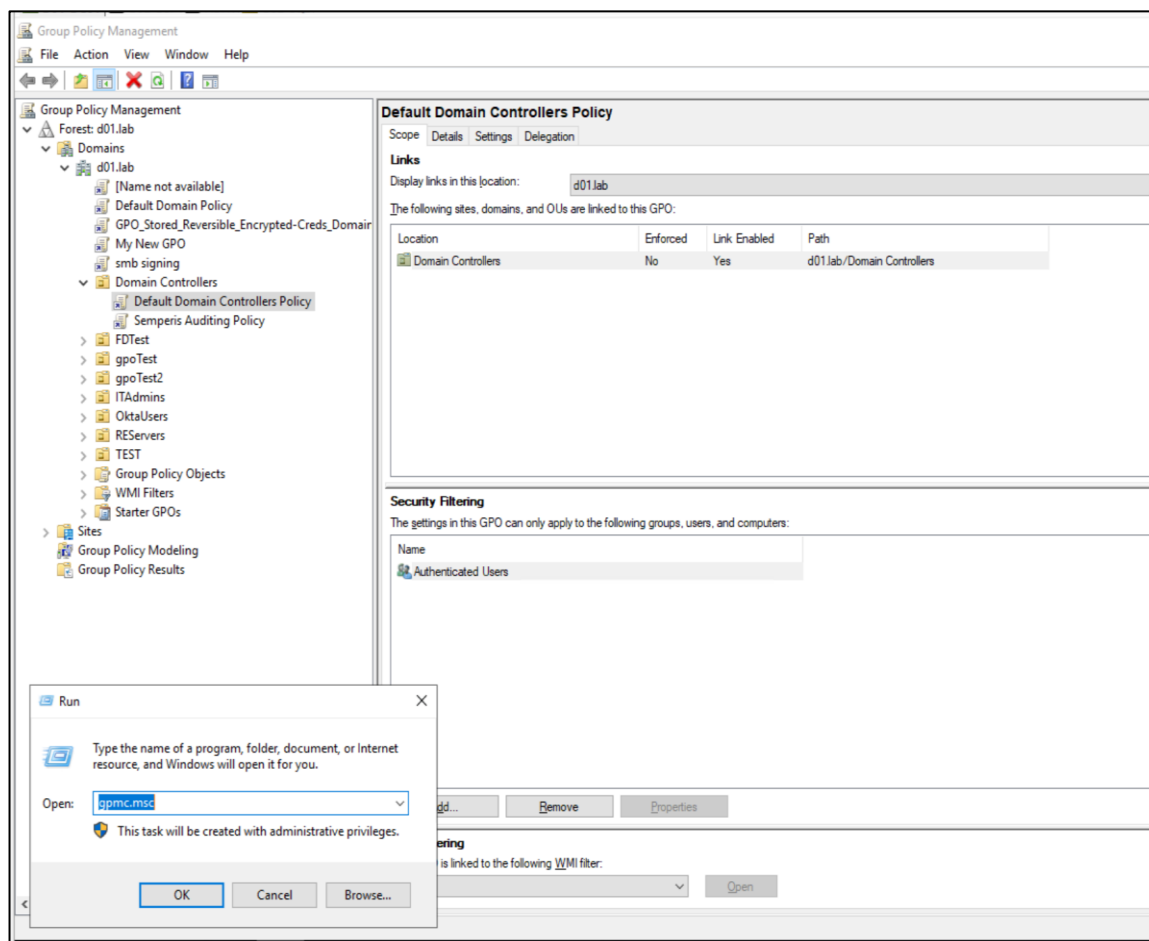




באמצעות הממשק אנחנו יכולים לערוך את ה-User Configuration וה-Configuration Computer מקומית על המחשב הזה. מה קורה אם אנו עורכים את המדיניות לוקאלית ואז מנהל הרשת שלנו מחיל עלינו מדיניות מנוגדות? נגלה בהמשך 😊

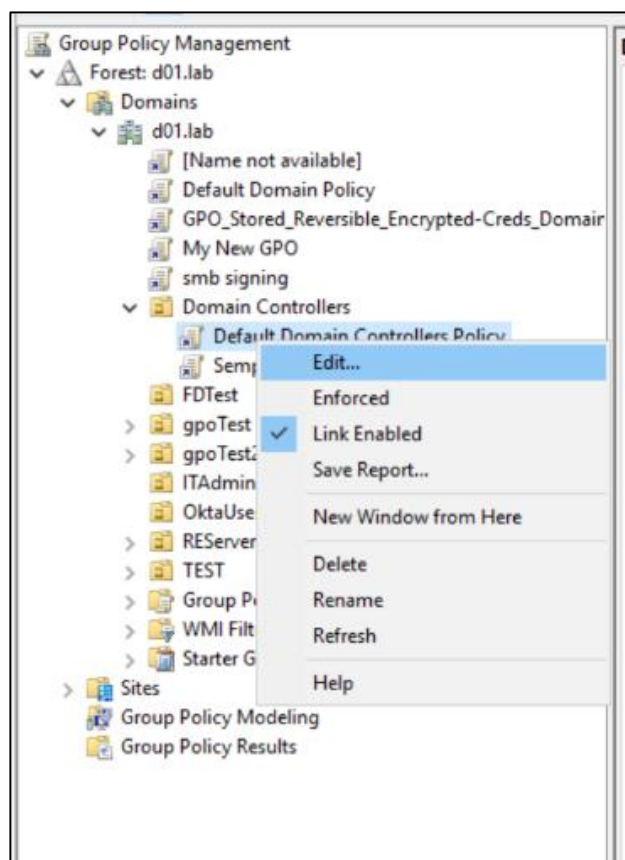
ההמלצה של מיקרוסופט היא לערוך את המדיניות באמצעות Group Policy Editor שמגיע כחלק מחבילת ה-Active Directory. בצורה זו, נוכל להחיל מדיניות לפי Scope-ים ספציפיים, על דומיין מסוים, OU מסוים וכו. נוכל לפתוח על ה-DC את ה-Group Policy Management console באמצעות כתיבה בשורת ה-Run:

```
gpmmc . msc
```



בעמוד זה, נוכל לראות את כל המדיניות בדומיין, ואת כל ה-Container-ים. אם מדיניות מוחלת על Container מסוים אנו נראה אותה תחתיו. למשל, נוכל לראות שתחת Domain Controllers יש 2 מדיניות.

על מנת לערוך מדיניות נלחץ עליה לחצן ימני -> Edit



היתרון בשימוש בכלי הדומיין להחלת מדיניות היא שנוכל בצורה קלה ומהירה להחיל חוקים על כלל האובייקטים בדומיין או על קבוצה רחבה של אובייקטים.

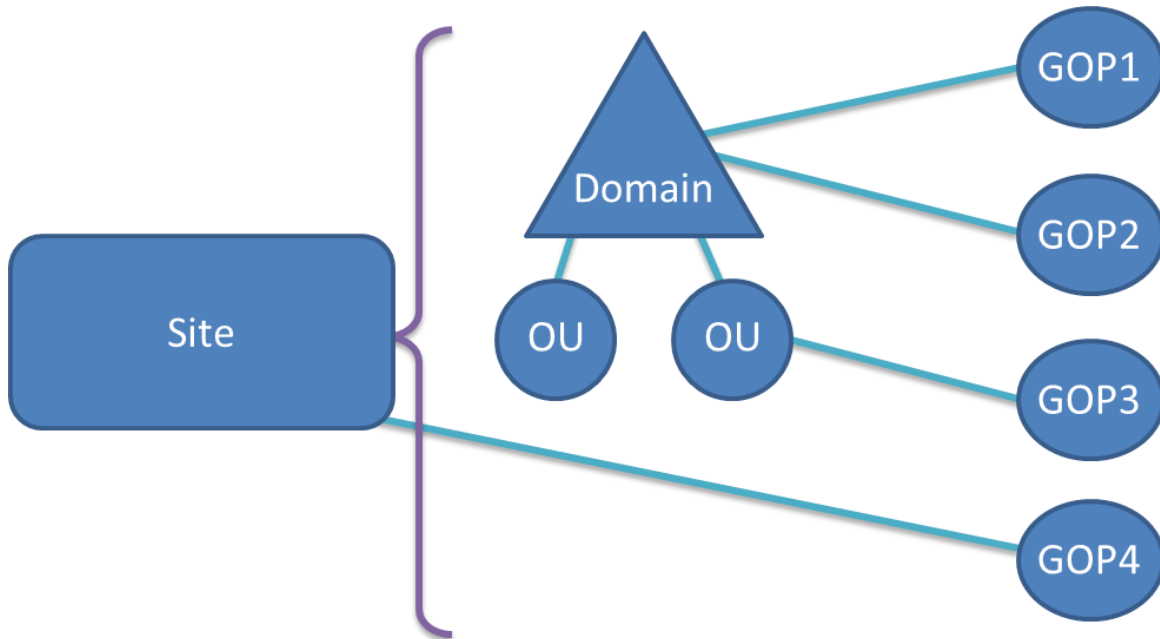
כמו כן, כלל המידע יהיה מאוחסן במקום אחד, ולא לוקאלית על כל מכונה, כך שיותר קל לנהל ולהכיר את מה שקורה בדומיין.

כמה דוגמאות של החלת מדיניות על קבוצות שונות:

- GPO שמוחל על Site יהיה מוחל על כל המשתמשים והמכונות באותו ה-Site
- GPO שמוחל על Domain יהיה מוחל על כלל המשתמשים והמכונות בדומיין, בירושה, על כלל המשתמשים והמכונות ב-OU-ים של הדומיין
- GPO המוחל על OU יהיה מוחל על כלל המשתמשים והמכונות תחת ה-OU, ובירושה, גם על כל Child OU



להלן תרשים המסביר המציג את התרחישים הנ"ל:



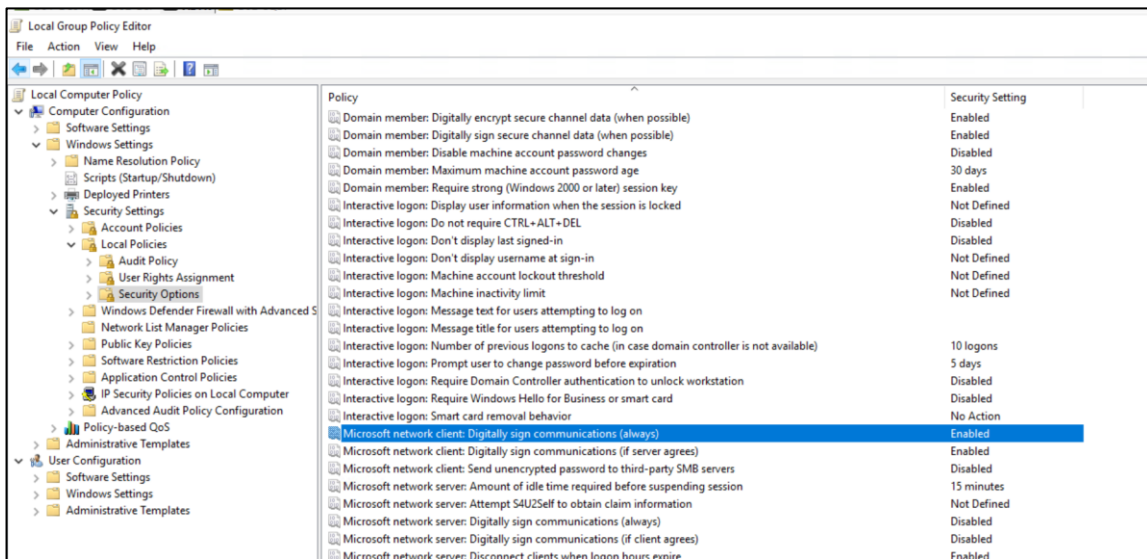
[Groyo Policy and the Active Directory, התרשים המקורי באתר של Microsoft מביך מדי מדי לשים אותו כאן]

נקודה חשובה ואחרונה לפני שאנחנו נכנסים למעמקי ה-GPO והיכולות שלו היא ההירכיה! אז מה קורה אם אני החלתי לוקאלית על המחשב שלי את המדיניות הבאה כי חשוב לי שהתקשורת שלי תהיה חתומה (אגב, מומלץ):

Microsoft network server: Digitally sign communications (always)

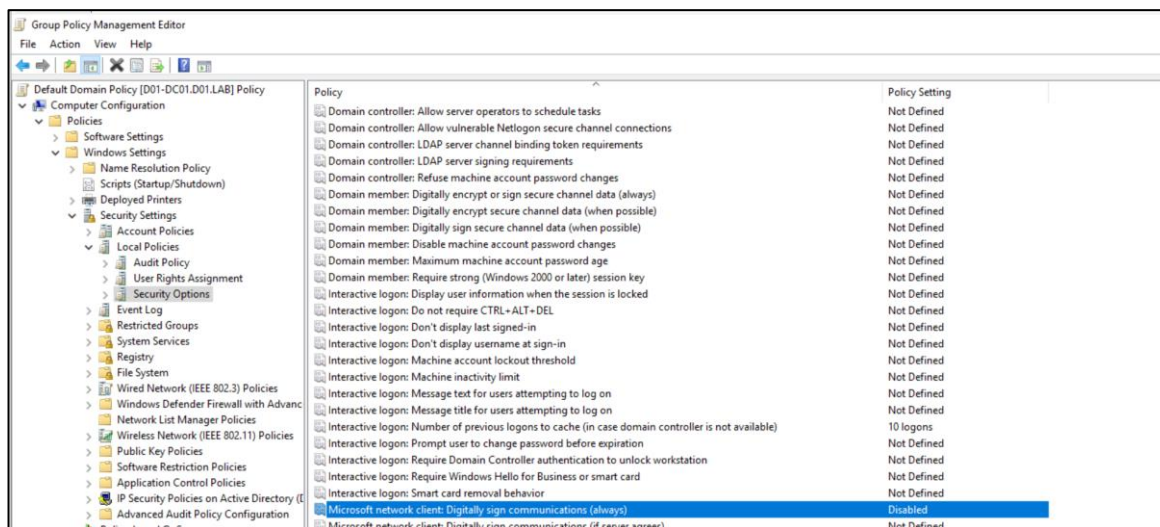
נוכל לראות את ההגדרה הזו כאן:

Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Options





אז כל הכבוד, אכפת לי מביטחון התקשורת שלי, אבל למנהל הרשת שלי פחות אכפת והוא הגדיר לכל הדומיין שההגדרה הזו היא Disabled:



לעומת המדיניות הקודמת שהיתה לוקאלית אנחנו יכולים לראות בממשק הניהול שהמדיניות הזו מוחלת על כל הדומיין!

היררכיית GPO

1. Local GPO
2. GPO linked to sites
3. GPO linked to Domain
4. GPO linked to OU (אם יש תתי OU-ים, קודם מוחלים ה-Parent ולאחר מכן ה-Child)

זאת אומרת, שהדבר הראשון שנדרס הוא המדיניות הלוקאלית, והדבר האחרון שנדרס (אז תכלס אף אחד לא יכול לדרוס) הוא המדיניות שמוחלת ישירות על ה-OU. אז במקרה שלנו, המדיניות הדומינית תדרוס את המדיניות הלוקאלית והתקשורת שלנו תישאר לא מאובטחת ☺

הדוגמא הזו נועדה להסביר כמה חשוב סדר החלת המדיניות, אם מנהל הרשת חושב שבכך שהחיל מדיניות על כל הדומיין, הדומיין מאובטח, הוא טועה. שכן אם יש מדיניות שמוחלות ספציפית על OU-ים הן ידרסו את המדיניות הדומינית. יש מגוון מתקפות מבוססות על היררכיית ה-GPO, ונגע בהן בהמשך.

אגב, היררכיה זו היא הבסיסית, אך ניתן להשתמש ב-**No override** על מנת לא לדרוס את המדיניות שלנו. אז אם במקרה שלנו מנהל הרשת ישתמש ב-**No override** על הפוליסת דומיין, היא לא תדרס על ידי המדיניות שמוגדרות על ה-OU-ים.



כמו כן, ניתן גם לחסום ירושה של מדיניות באמצעות **Block inheritance** כדי למנוע ממדיניות להיות מוחלות על תתי OU-ים.

- **No Override** תמיד תדרוס **Block inheritance**
- מדיניות לוקאלית לא יכולה להשתמש באופציה של **Block inheritance** או **No Override**

אז ראינו שיש דבר כזה GPO, מדיניות חוקים שנוכל להחיל לוקאלית או דומיינית. כמו כן, ראינו שיש 2 סוגים של פלטפורמות חוקים בתוך כל מדיניות:

1. User Configuration - מדיניות שמוחלות על משתמש
2. Computer Configuration - מדיניות שמוחלות על מכונה

יש הרבה מאוד סוגים של חוקים שניתן להחיל, אנחנו נתמקד רק במספר דיי מצומם שכן כמות האופציות כאן היא ענקית.

משיכת המדיניות

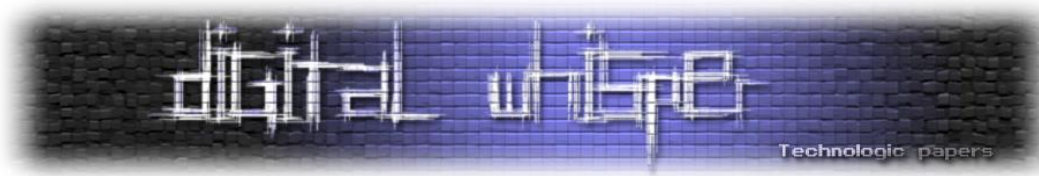
הדבר האחרון שאנחנו צריכים לפני שנצלול פנימה, הוא לענות על שאלה חשובה, איך מחשב או משתמש יודע מה המדיניות שלו?? מאיפה המחשב לוקח את המדיניות ומחיל אותן על עצמו?

אה! בואו נראה! כלל המדיניות נמצאות בתקיה על ה-DC שנגישה לכלל הדומיין בשם sysvol:

| Name | Date modified | Type | Size |
|--|-------------------|-------------|------|
| {00BF7438-F282-4587-9D2E-663887978637} | 1/24/2023 3:00 PM | File folder | |
| {0A0D8D07-7716-4CDE-BA74-55693CC26DCF} | 1/24/2023 3:02 PM | File folder | |
| {0AC9FA00-3808-4117-8FAD-7EA9B33AA299} | 1/24/2023 3:01 PM | File folder | |
| {0B18E105-1B7E-4F1F-9547-07A7D50D81E5} | 1/24/2023 3:01 PM | File folder | |
| {0B708FC2-DB0E-43D2-8BBB-7FDB8A9E0C5E} | 1/24/2023 3:02 PM | File folder | |
| {0B8367FA-04D8-4647-B5C0-78C4428BB272} | 1/23/2023 9:39 AM | File folder | |
| {0BC8EA19-8967-44A3-94E9-87C5259E92FF} | 1/24/2023 3:00 PM | File folder | |
| {0BF3BE6A-A25A-417B-BD08-632179D7B5D9} | 1/24/2023 3:02 PM | File folder | |
| {0BF6055A-A121-4B57-84FC-2A80BFD25A5B} | 1/24/2023 3:00 PM | File folder | |
| {0CDCC8B2-A9FE-4D8F-A251-F7BF462C01E7} | 1/24/2023 3:00 PM | File folder | |
| {0DD868D3-9AA3-475E-88CE-E00556443F3E} | 1/24/2023 3:00 PM | File folder | |
| {0F8F4837-D039-4A67-B676-23FAE7A5B053} | 1/24/2023 3:00 PM | File folder | |
| {01E75D2D-C014-4F18-80F3-3EA67C775C58} | 1/24/2023 3:00 PM | File folder | |
| {1A914467-539A-4DF9-A5D7-AD73123FF808} | 1/23/2023 9:39 AM | File folder | |
| {1B12BF20-4E80-4838-8ADA-7C6ADE0E992} | 1/24/2023 3:02 PM | File folder | |
| {1B6555A1-75B4-4572-AF94-508C3C394BF7} | 1/24/2023 3:01 PM | File folder | |
| {1B374645-ECC7-4A3F-811F-028D106FD22A} | 1/23/2023 9:39 AM | File folder | |
| {1BC1DFCF-C112-4650-A41F-43DC67CB4BCD} | 1/24/2023 3:01 PM | File folder | |
| {1BDD4FA5-3676-4129-BE54-FCFA2448CC4D} | 1/23/2023 9:39 AM | File folder | |

כל מחשב בדומיין פונה לנתיב הזה באחד משני המקרים:

1. המחשב נדלק (פונה למשור Computer configuration)
2. משתמש מתחבר (פונה למשור User configuration)



בתוך כל תקיית מדיניות נראה בערך משהו כזה:

| Name | Date modified | Type | Size |
|---------|-------------------|-----------------------|------|
| Machine | 1/24/2023 3:01 PM | File folder | |
| User | 1/24/2023 3:01 PM | File folder | |
| GPT.INI | 1/24/2023 3:01 PM | Configuration sett... | 1 KB |

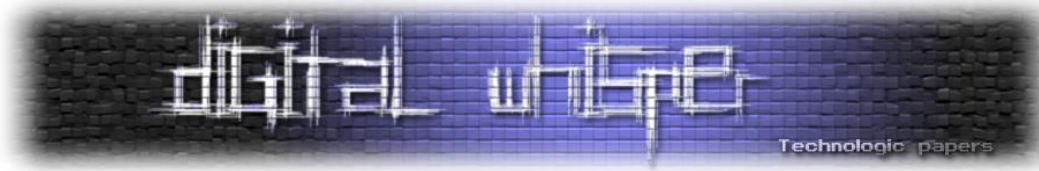
במידה ויש הגדרות במדיניות הן יאוחסנו במספר קבצים שונים (כמו למשל הקובץ GPT.INI שניתן לראות) ואנחנו נדבר עליהם בהמשך כדי להבין איך ניתן לכתוב detection rules ל-GPO.

נציץ רק לרגע בדוגמא: אחד הקבצים המחזיקים את התוכן של המדיניות נקרא GptTmpl.inf. בקובץ זה נמצא את מרבית הגדרות ה Security של המדיניות. הוא בדרך כלל נמצא במיקום הזה:

```
\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Microsoft\Windows NT\SecEdit\GptTmpl.inf
```

בואו נראה את התוכן שלו:

```
1 [Unicode]
2   Unicode=yes
3 [System Access]
4   MinimumPasswordAge = 1
5   MaximumPasswordAge = 42
6   MinimumPasswordLength = 7
7   PasswordComplexity = 1
8   PasswordHistorySize = 24
9   LockoutBadCount = 0
10  RequireLogonToChangePassword = 0
11  ForceLogoffWhenHourExpire = 0
12  ClearTextPassword = 0
13  LSAAnonymousNameLookup = 0
14 [Kerberos Policy]
15  MaxTicketAge = 10
16  MaxRenewAge = 7
17  MaxServiceAge = 600
18  MaxClockSkew = 5
19  TicketValidateClient = 1
20 [Version]
21  signature="CHICAGO"
22  Revision=1
23 [Privilege Rights]
24  SetCbPrivilege = normal
25 [Registry Values]
26  MACHINE\System\CurrentControlSet\Control\Lsa\NoLMHash=4,0
27  MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\RequireSecuritySignature=4,1
28  MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\CachedLogonsCount=1,"10"
```

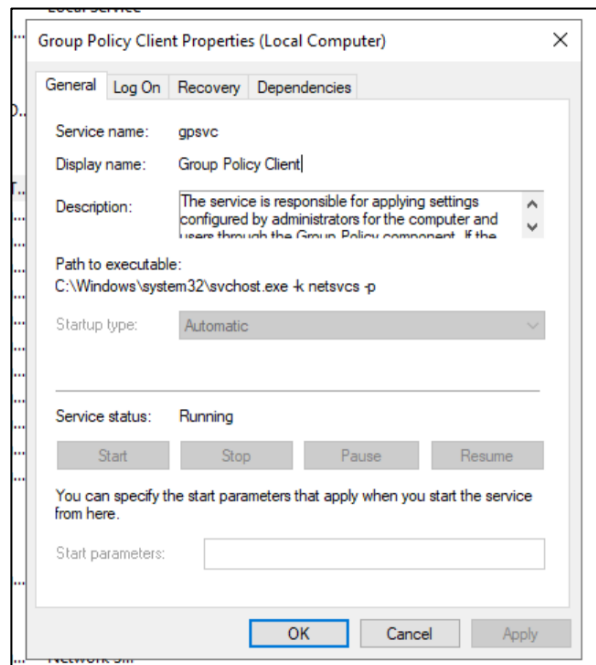


נוכל לראות פה כל מיני הגדרות תחת קטוגריות שונות כמו:

- נתיבי Registry שאחראים על קונפיגורציות אבטחתיות שונות (כמו למשל RequireSecuritySignature שקינפגנו ממש לפני כמה עמודים!)
- Privilege Rights שמיד נדבר עליו
- Kerberos Policy - שמחיל קונפיגורציה על תוחלת כרטיס למשל.
- System Access - שמחיל הרבה הגדרות סיסמא כמו אורך מינימלי, זמן חיים וכו.

בהמשך, אציג מספר סקריפטים שיודעים לפרסר את הקבצים האלו לצורך כתיבת חוקים ☺

ואיך המחשב יודע אילו מדיניות מוחלות עליו? באמצעות ה-Service החביב הזה, GPClient יודע לתשאל את הדומיין כדי לקבל את כל המידע הנחוץ.



דוגמא לשאילתת LDAP על המדיניות שמתבצעת על ידי GPClient:

| | | | | | |
|--------------|------|------|------------------|----------|---|
| 192.168.0.2 | LDAP | 179 | SASL | GSS-API | Integrity: searchResEntry(4005) "CN=D01,CN=Sites,CN=Configuration,DC=d01,DC=lab" searchResDone(4005) success [2 results] |
| 192.168.0.11 | LDAP | 1149 | SASL | GSS-API | Integrity: searchRequest(4006) "cn=policies,cn=system,DC=d01,DC=lab" wholeSubtree |
| 192.168.0.2 | LDAP | 8371 | SASL | GSS-API | Integrity: searchResEntry(4006) "CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Polices,CN=System,DC=d01,DC=lab" searchResEntry(4006) |
| 192.168.0.21 | LDAP | 1952 | bindRequest(11) | "<ROOT>" | sasl |
| 192.168.0.2 | LDAP | 266 | bindResponse(11) | success | |

```

> Filter: (gpUserExtensionNames=[*])
v Filter: (|(distinguishedName=cn={331A6179-32FC-4A0D-B808-8112990E912A},cn=policies,cn=system,DC=d01,DC=lab)(distinguishedName=cn={331A6179-32FC-4A0D-B808-8112990E912A},cn=policies,cn=system,DC=d01,DC=lab)(distinguishedName=cn={331A6179-32FC-4A0D-B808-8112990E912A},cn=policies,cn=system,DC=d01,DC=lab))
  and item: or (1)
    or: (|(distinguishedName=cn={331A6179-32FC-4A0D-B808-8112990E912A},cn=policies,cn=system,DC=d01,DC=lab)(distinguishedName=cn={331A6179-32FC-4A0D-B808-8112990E912A},cn=policies,cn=system,DC=d01,DC=lab))
      or: 6 items
        > Filter: (distinguishedName=cn={331A6179-32FC-4A0D-B808-8112990E912A},cn=policies,cn=system,DC=d01,DC=lab)
        > Filter: (distinguishedName=cn={C1C5FDEF-8C17-4C06-94DB-5306A0549026},cn=policies,cn=system,DC=d01,DC=lab)
        > Filter: (distinguishedName=cn={9879C8E3-EB4B-4103-8E42-C83EFAC7A7B2},cn=policies,cn=system,DC=d01,DC=lab)
        > Filter: (distinguishedName=cn={053DF8AB-6C9C-4335-BA26-7727D33FA1A4},cn=policies,cn=system,DC=d01,DC=lab)
        > Filter: (distinguishedName=cn={7cdd0219-58a2-4c25-bee7-35e0f0e87d85},cn=policies,cn=system,DC=d01,DC=lab)
        > Filter: (distinguishedName=CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Polices,CN=System,DC=d01,DC=lab)
  items: 11 items
  
```




דוגמא לפקטת SMB של משיכת המדיניות שמתבצעת על ידי GPClient:

```
518 Create Request File: d01.lab\Policies\{9879C8E3-EB4B-4103-8E42-C83EFAC7A7B2}\User\Scripts\Scripts.ini
410 Create Response File: d01.lab\Policies\{9879C8E3-EB4B-4103-8E42-C83EFAC7A7B2}\User\Scripts\Scripts.ini
146 Close Request File: d01.lab\Policies\{9879C8E3-EB4B-4103-8E42-C83EFAC7A7B2}\User\Scripts\Scripts.ini
```

נוכל לראות בתמונה ממש את התוכן של הקובץ scripts.ini:

```
[.L.o.g.o.n.].
.O.C.m.d.L.i.n.e.=.n.e.w...b.a.t.
.O.P.a.r.a.m.e.t.e.r.s.=.
...p.SMB@.....
...8.....8 4...B;
```

אם אנחנו רוצים לראות אילו מדיניות מוחלות על המכונה שלנו, נוכל להשתמש בפקודה:

```
gpresult /R
```

דוגמא לחלק מהפלט:

```
Microsoft (R) Windows (R) Operating System Group Policy Result tool v2.0
© 2018 Microsoft Corporation. All rights reserved.

Created on [ 7/5/2023 at 9:19:47 AM

RSOP data for d01\d01admin on D01-DC01 : Logging Mode
-----

OS Configuration:          Primary Domain Controller
OS Version:                 10.0.17763
Site Name:                  D01
Roaming Profile:            N/A
Local Profile:              C:\Users\Administrator
Connected over a slow link?: No

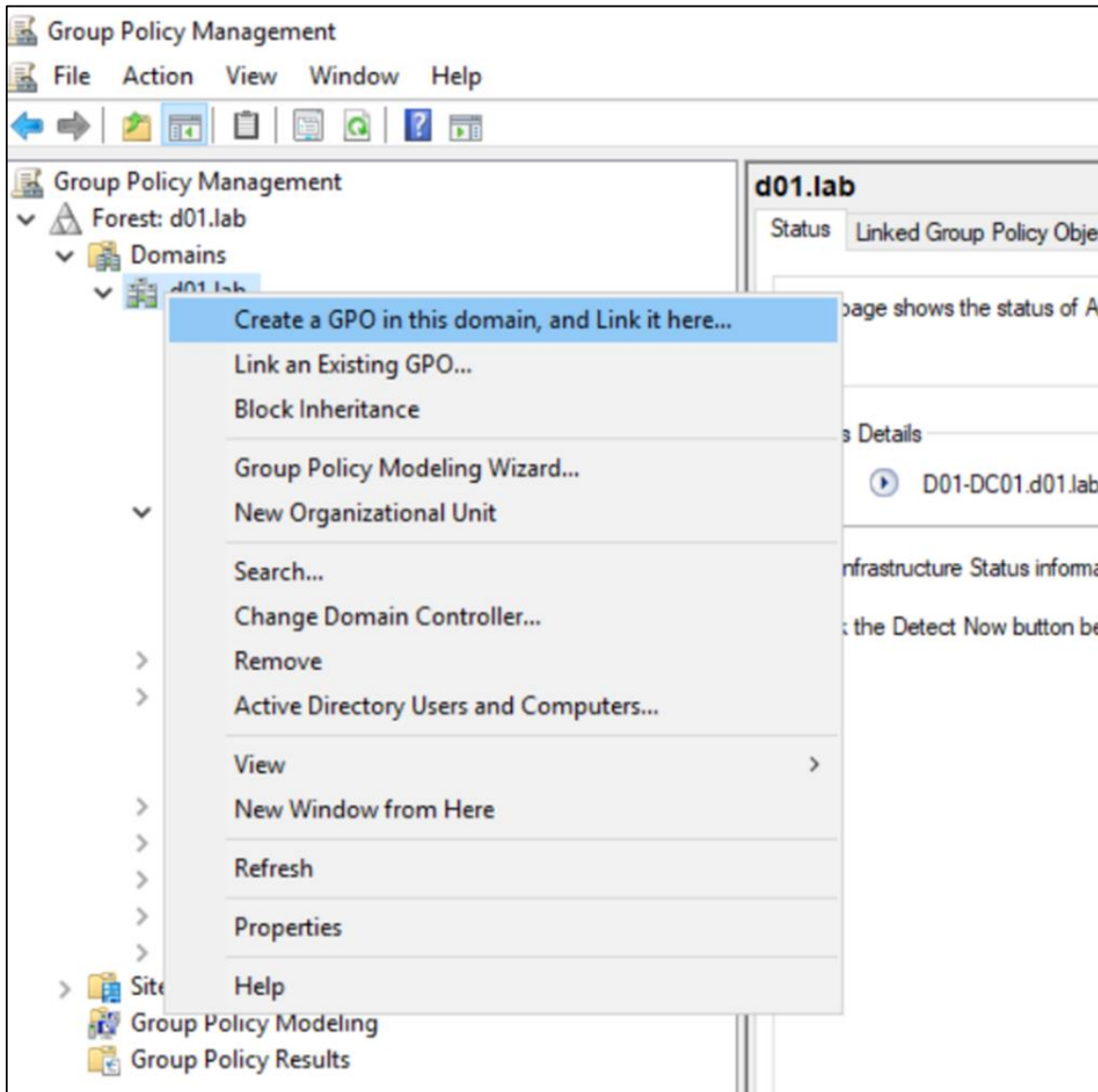
COMPUTER SETTINGS
-----
CN=D01-DC01,OU=Domain Controllers,DC=d01,DC=lab
Last time Group Policy was applied: 7/5/2023 at 9:15:41 AM
Group Policy was applied from:    D01-DC01.d01.lab
Group Policy slow link threshold: 500 kbps
Domain Name:                     d01
Domain Type:                     Windows 2008 or later

Applied Group Policy Objects
-----
Default Domain Controllers Policy
Semperis Auditing Policy
Default Domain Policy
My New GPO
smb signing
```

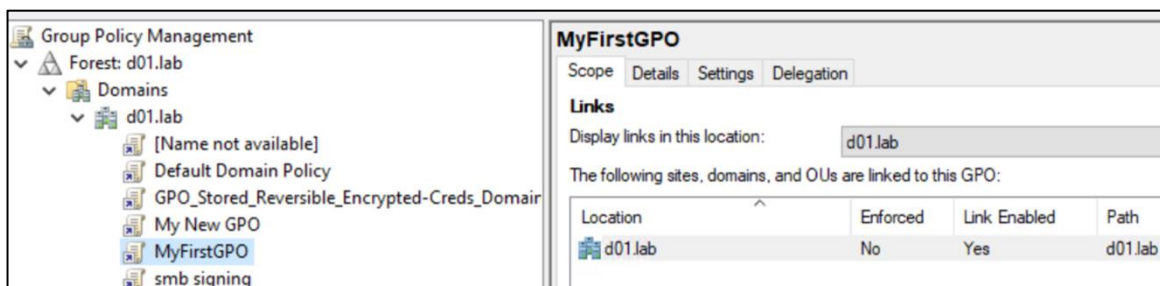


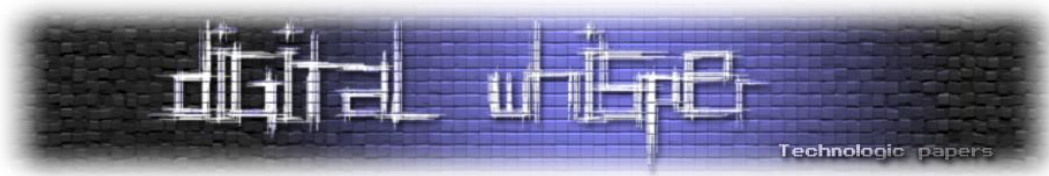
בואו לבנות מדיניות!

בבניית מדיניות דומיינית ובה נגדיר חוק אחד מ-User Configuration ואחד מ-Computer Configuration. בקונסול הניהול, נלחץ על לחצן ימני ונבחר ליצור מדיניות חדשה שתקושר לדומיין:

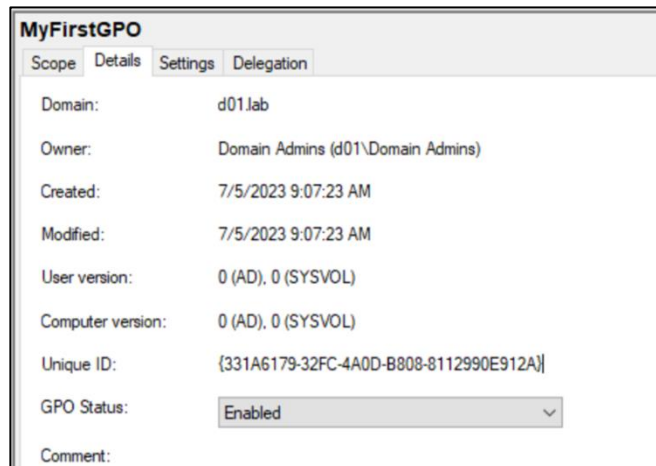


ניתן לה שם חביב ונוכל לראות אותה כעת בין המדיניות שלנו:





תחת החלונית Details נוכל גם לראות את ה-GUID של המדיניות ועוד קצת מידע עליה:



כעת, נלחץ לחצן ימני על המדיניות ונבחר Edit. אנחנו ניצור 2 חוקים:

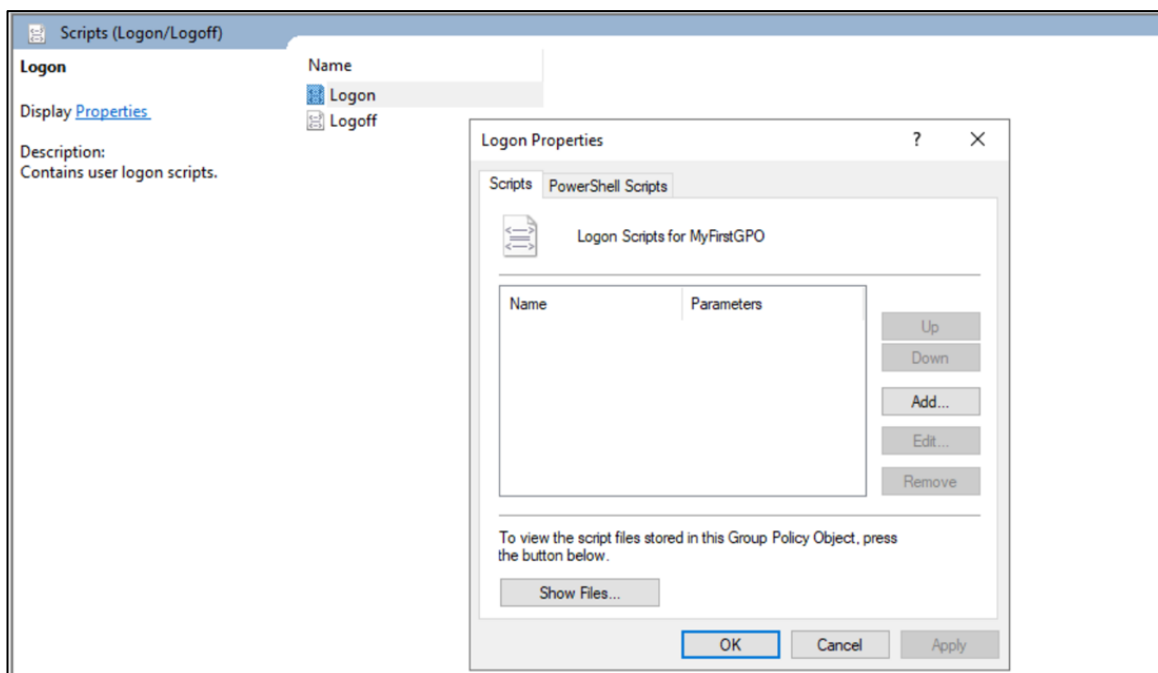
1. User logon script
2. Computer user right

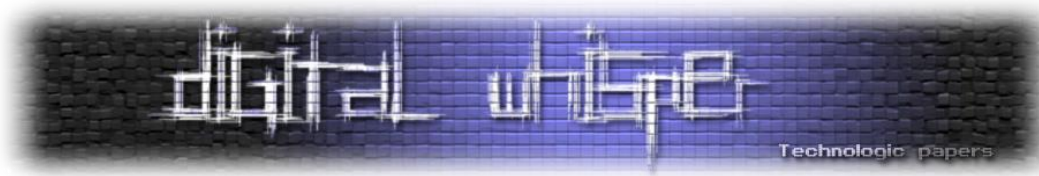
עם שני החוקים האלה ניתן להשיג אחיזה על כלל הדומיין, אז אל תמעיטו בערכם של GPO!

User Logon Script

נתחיל מהראשון, סביר להניח שרובכם מכירים את הקונספט של logon script, אך למי שלא: מדובר בסקריפט שירוץ בכל פעם שהמשתמש יתחבר למחשב. אז כדי ליצור logon script נלך לנתיב הבא:

User Configuration\Policies\Windows Settings\Scripts (Logon/Logoff)

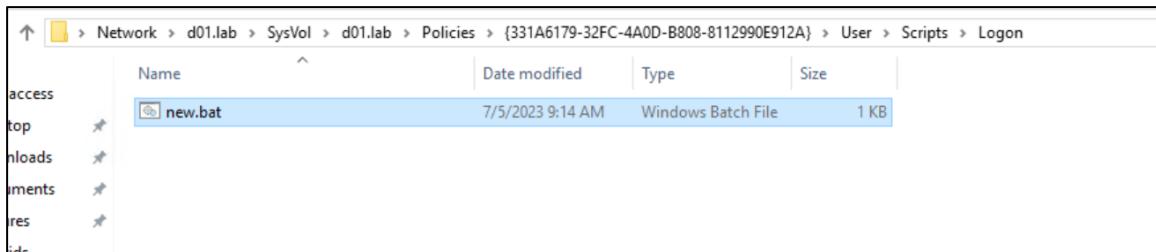




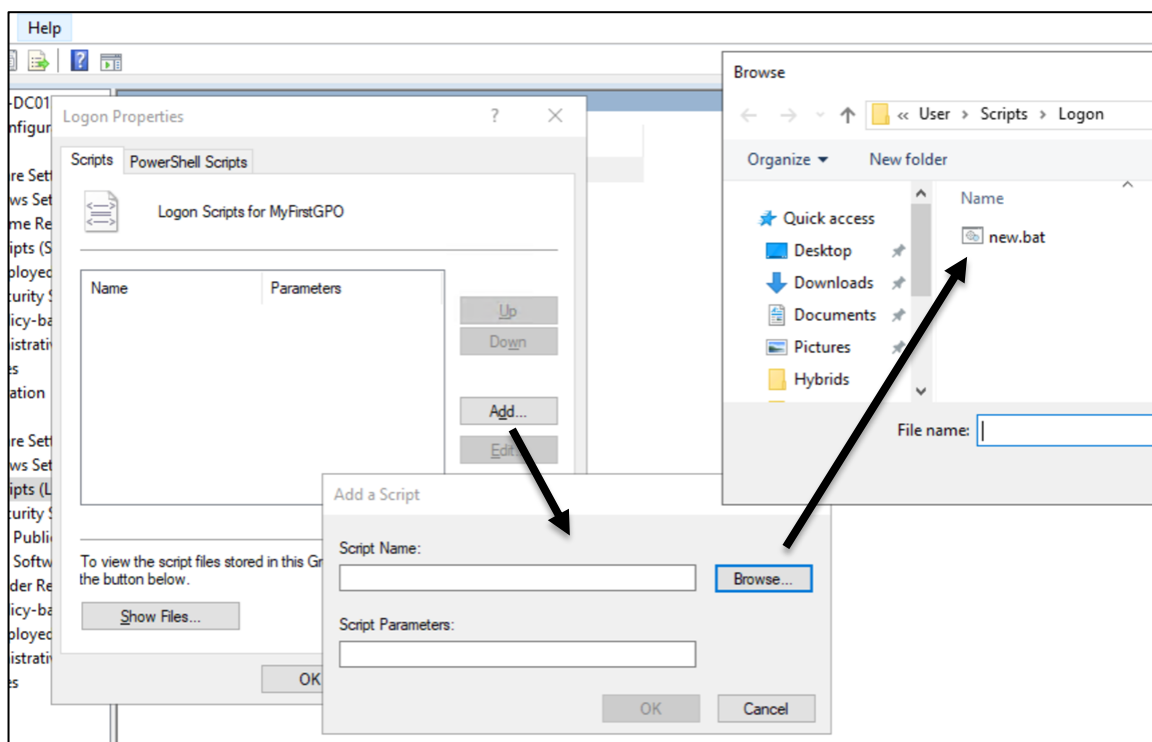
כעת, אנו צריכים לספק סקריפט, נכתוב סקריפט Bat קצר שיקפיץ לנו pop! כתבו בתוך קובץ עם סיומת bat השורה הבאה:

```
msg * "Hello World"!
```

נמקם את הסקריפט בנתיב של המדיניות, תחת הנתיב: User/Scripts/logon



כעת נלחץ על add -> browse -> הסקריפט שלנו:

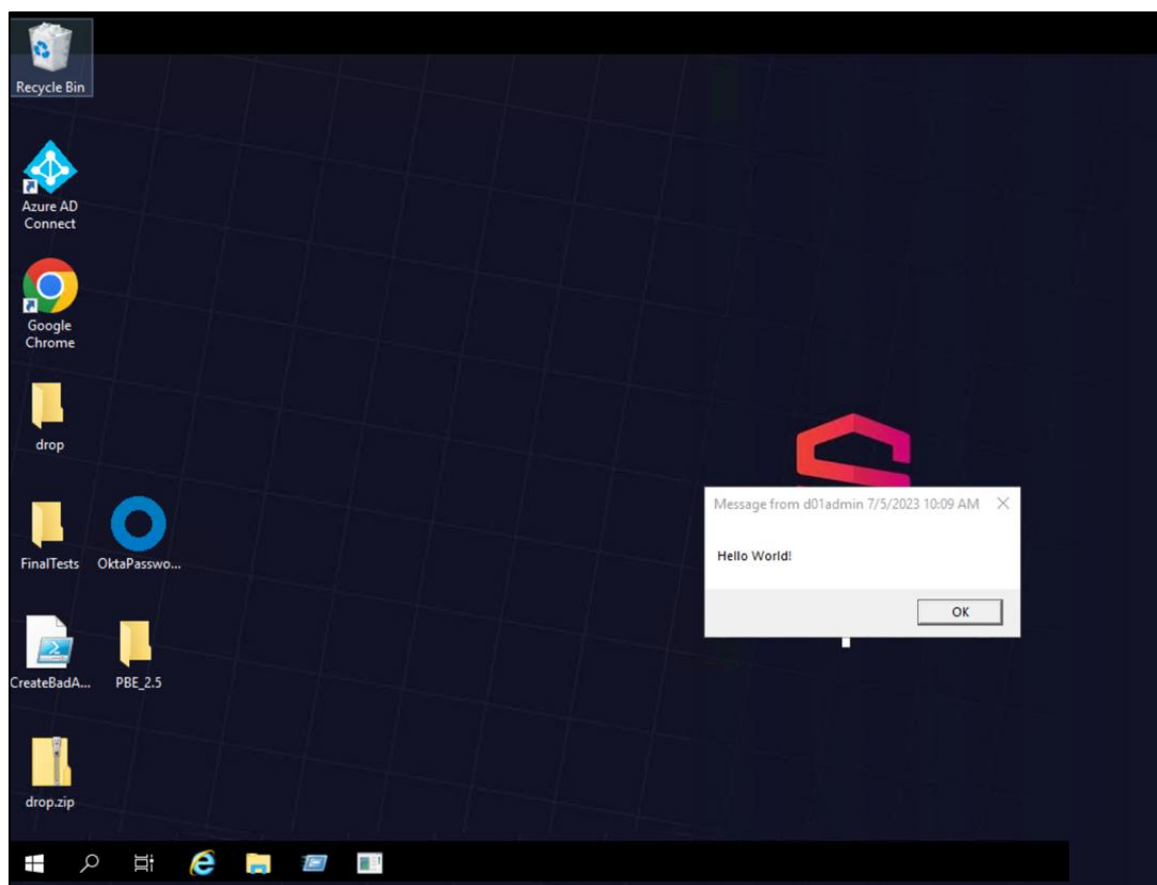


אין לסקריפט שלנו פרמטרים, לכן סיימנו. כעת, נצטרך שה-GPO יתעדכנו. נוכל לחכות או פשוט להריץ על העמדה הרצויה:

```
gpupdate /force
```

שיעדכן לנו מיד את ההגדרות.

בואו נתחבר עם משתמש חדש ונראה מה קורה!



אז בעצם יצרנו User logon script בתוך ה-GPO שלנו שמוחל על כל הדומיין, וככה הרצנו קוד על כל עמדה בדומיין שיתבצע אליה login.

Computer User Rights

כעת נעבור ל-Computer Configuration, אנחנו הולכים להגדיר User right. לא הרבה מכירים את הקונספט, אך תחת הנתיה הבא:

Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights

Assignment



נוכל לתת הרשאות מאוד מעניינות למשתמשים בדומיין, בואו נסתכל מה יש שם:

| Policy | Policy Setting |
|---|----------------|
| Access Credential Manager as a trusted caller | Not Defined |
| Access this computer from the network | Not Defined |
| Act as part of the operating system | normal1 |
| Add workstations to domain | Not Defined |
| Adjust memory quotas for a process | Not Defined |
| Allow log on locally | Not Defined |
| Allow log on through Remote Desktop Services | Not Defined |
| Back up files and directories | Not Defined |
| Bypass traverse checking | Not Defined |
| Change the system time | Not Defined |
| Change the time zone | Not Defined |
| Create a pagefile | Not Defined |
| Create a token object | Not Defined |
| Create global objects | Not Defined |
| Create permanent shared objects | Not Defined |
| Create symbolic links | Not Defined |
| Debug programs | Not Defined |
| Deny access to this computer from the network | Not Defined |
| Deny log on as a batch job | Not Defined |
| Deny log on as a service | Not Defined |
| Deny log on locally | Not Defined |
| Deny log on through Remote Desktop Services | Not Defined |
| Enable computer and user accounts to be trusted for delega... | Not Defined |
| Force shutdown from a remote system | Not Defined |
| Generate security audits | Not Defined |
| Impersonate a client after authentication | Not Defined |
| Increase a process working set | Not Defined |
| Increase scheduling priority | Not Defined |
| Load and unload device drivers | Not Defined |
| Lock pages in memory | Not Defined |
| Log on as a batch job | Not Defined |
| Log on as a service | Not Defined |
| Manage auditing and security log | Not Defined |
| Modify an object label | Not Defined |
| Modify firmware environment values | Not Defined |
| Obtain an impersonation token for another user in the same... | Not Defined |
| Perform volume maintenance tasks | Not Defined |
| Profile single process | Not Defined |
| Profile system performance | Not Defined |

כפי שניתן לראות, יש מספר הרשאות חזקות בצורה קיצונית כגון:

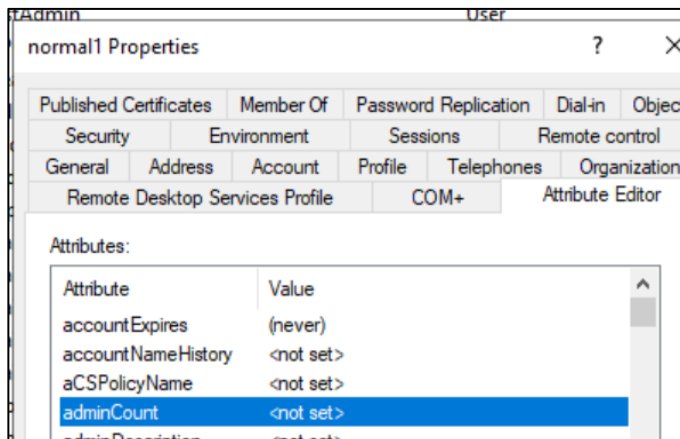
- Act as part of the operating system (או כפי שרבים מכנים אותה - SeTcbPrivilege)
 - Debug programs או בשם המוצלח יותר - SeDebugPrivilege
 - Impersonate a client after authentication או SeImpersonatePrivilege
- ולא חסרים עוד...

נקודות מעניינות:

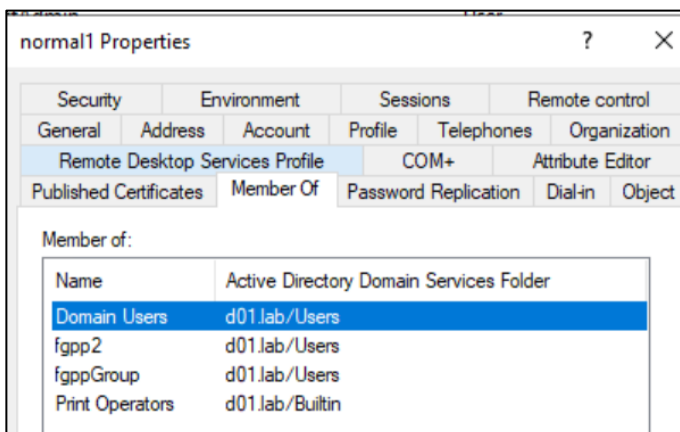
- כאשר אנחנו מחליטים מי יהיו המשתמשים החזקים שלנו ברשת, האם אנחנו מודעים למשתמשים האלו? שיכולים להשיג אחיזה בכל הדומיין? סביר להניח שלא...
- האם הייתי חושד במשתמש הזה? האם הייתי מצליח באמצעות מערכת ניטור להבין שמדובר במשתמש חזק מאוד?

איך נראות הגדרות של משתמש כזה? בואו נסתכל!

בתמונה מתחת אתם יכולים לראות שנתתי למשתמש normal1 את ההגדרה SeTcbPrivilege, זאת אומרת שהמשתמש הזה בפועל הוא בעל הרשאות מאוד חזקות ואנחנו יודעים את זה דרך המדיניות. אין למשתמש admincount - לא נוכל להשתמש בשדה זה כדי להבין שמדובר במשתמש חזק:



המשתמש לא חבר בקבוצה בעלת הרשאות גבוהות:

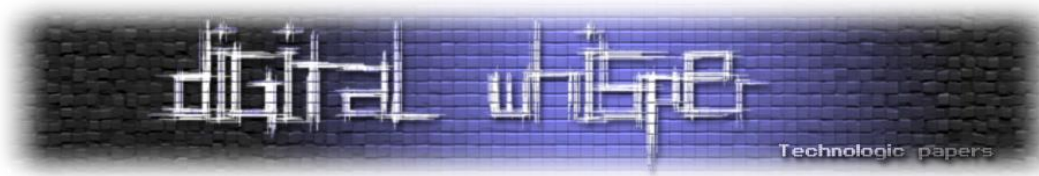


אך אם נריץ /priv /whoami מ-elevated context נוכל לראות את ההרשאות הבאות:

```
C:\Windows\system32>whoami & whoami /priv
d01\normal1

PRIVILEGES INFORMATION
-----
Privilege Name            Description                    State
-----
SeMachineAccountPrivilege Add workstations to domain    Disabled
SeTcbPrivilege            Act as part of the operating system Disabled
SeLoadDriverPrivilege    Load and unload device drivers Disabled
SeShutdownPrivilege      Shut down the system          Disabled
SeChangeNotifyPrivilege  Bypass traverse checking      Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled

C:\Windows\system32>
```

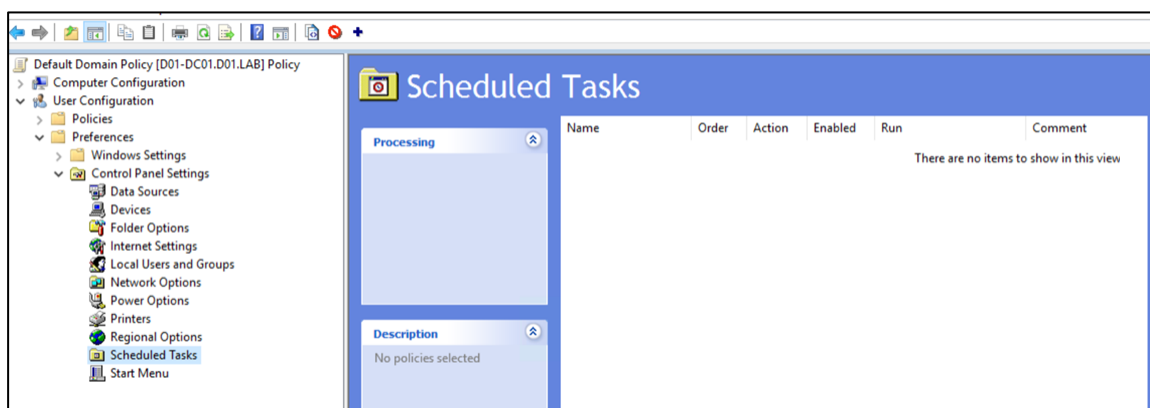


המסקנה: יכול להיות שיש לנו משתמשים בעלי הרשאות מאוד גבוהות בסביבה שאנחנו כלל לא מודעים אליהם (ואולי מסיבה זו הם גם פחות מוגנים), תוקף יכול להשתמש במשתמשים כאלה כדי להסלים הרשאות ולחילופין יכול ליצור משתמשים כאלו לצורך Backdoor.

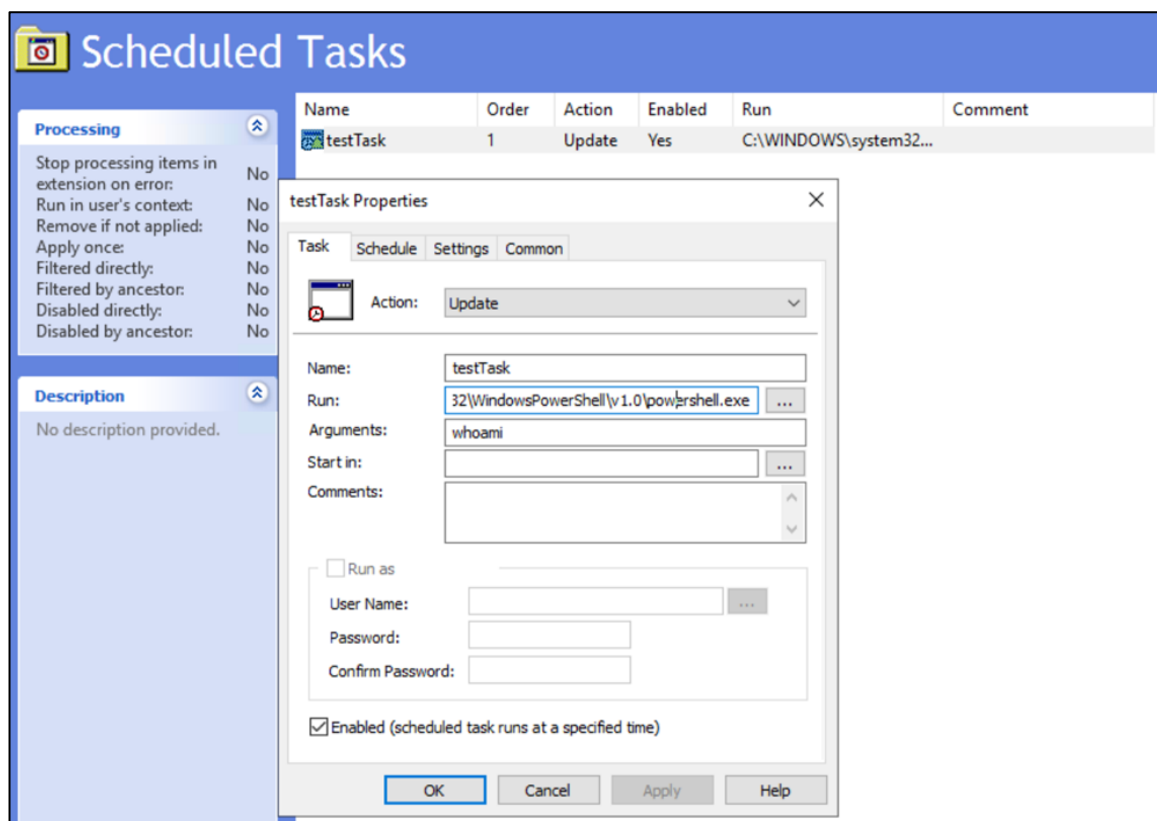
משימות מתוזמנות

לא אכיר במילים, אבל חשוב לציין שניתן ליצור משימות מתוזמנות דרך GPO. תחת התיב הבא נוכל לראות ממשק יפיה לקינפוג משימות מתוזמנות דרך מדיניות GPO משנת תרפ"ו:

User Configuration -> Preferences -> Control Panel Settings -> Scheduled Tasks



ניצור משימה מתוזמנת ובבין מה היכולות של הפעולה הזו:





תחת הנתיב הבא נוכל לראות את המשימה המתוזמנת שלנו:

```
\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\USER\Preferences\ScheduledTasks\ScheduledTasks.xml
```

תוכן הקובץ:

```
<?xml version="1.0" encoding="utf-8"?>
<ScheduledTasks clsid="{CC63F200-7309-4ba0-B154-A71CD118DBCC}"><Task clsid="{2DEECB1C-261F-4e13-9B21-16FB83BC03BD}" name="testTask"
image="2" changed="2023-07-05 11:53:18" uid="{67D747A1-DEE3-4F86-87D4-14E339BEDEB9}" userContext="0" removePolicy="0">
<Properties deleteWhenDone="0" startOnlyIfIdle="0" stopOnIdleEnd="0" noStartIfOnBatteries="1" stopIfGoingOnBatteries="1"
systemRequired="0" action="U" name="testTask" appName="C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe" args="whoami"
startIn="" comment="" enabled="1"><Triggers><Trigger type="DAILY" startHour="09" startMinutes="00" beginYear="2023" beginMonth="7"
beginDay="5" hasEndDate="0" repeatTask="0" interval="1"/></Triggers></Properties></Task>
</ScheduledTasks>
```

אז אנחנו יודעים שאנחנו יכולים לגרום למשימה מתוזמנת לרוץ על כל הדומיין, נחמד. נוכל לפרסר את הקבצים האלו כדי לקבל מידע על המשימות המתוזמנות שרצות לנו ברשת ☺

אז בואו נסכם מה למדנו עד כה

1. מה זה GPO
2. איך מחילים GPO על אובייקט
3. היררכיית ההחלה של GPO
4. Computer and User configuration
5. איך רואים מידע על ה-GPO שמוחלים עלי (gpresult)
6. איפה נמצאים הקבצים של ה-GPO ואיך המחשב קורא אותם
7. סוגי הגדרות שניתן להחיל כמו נתיבי registry, logon scripts, User Rights ומשימות מתוזמנות. אצרך רשימה של User Rights מעניינים בסוף המאמר ☺

פורסמו מספר חולשות ב-GPO ולפי מה שראיתי רובן מתבססות על אותו הקונספט - ניצול של ה-GPClient (שמופעל כאשר קוראים ל-GPupdate) על מנת לגרום לו להריץ בשבילנו קוד ולהשיג הסלמת הרשאות.

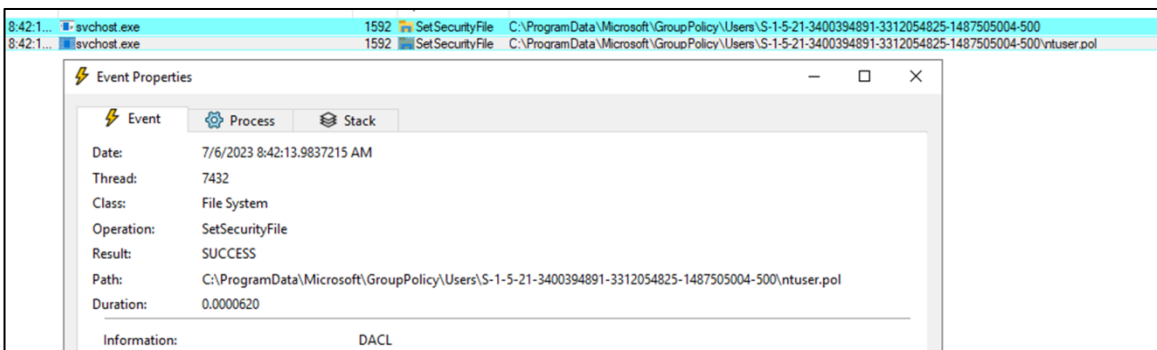
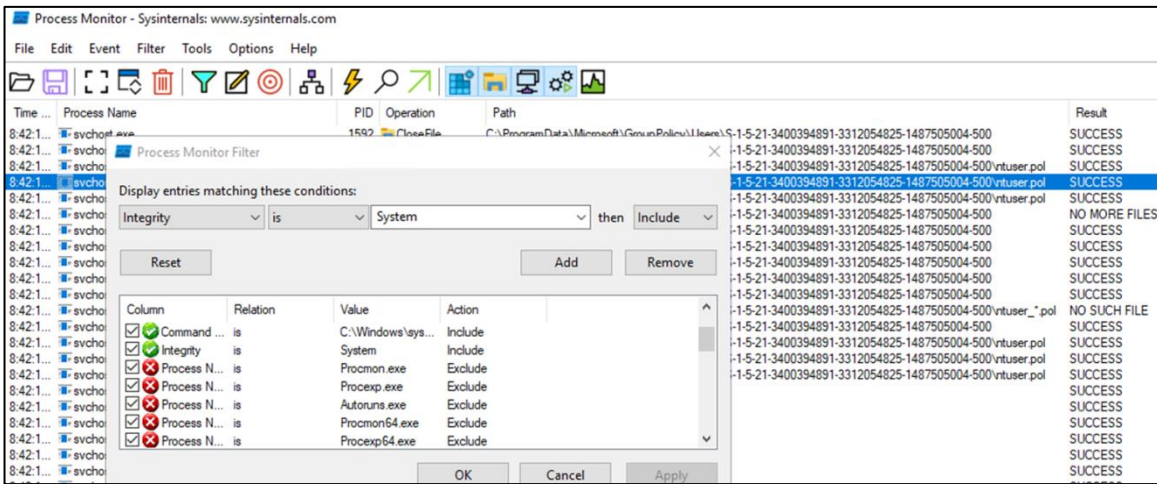
החולשה הזו מגיעה אלינו מבחור מקסים בשם Nabeel Ahmed. אני לא אסביר את כל תהליך מציאת החולשה, שכן מטרת הסעיף הזה היא לחשוף את הקורא לרעיון הכללי של חולשות במנגנון ה-GPO. כמובן שהיא גם מפוצ'פצ'ת כבר ולכן נאלצתי לקחת חלק מהתמונות מהמאמר עצמו, שכן החולשה לא ברת מימוש יותר. (קישור בסוף המאמר).

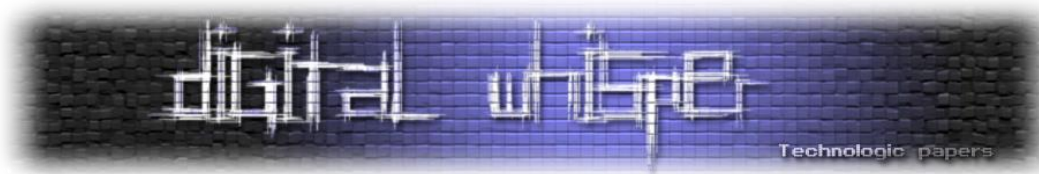
אז נתחיל, כאשר מתבצע עדכון ל-GPO, המדיניות נשמרות ב-Cache לוקאלי. ספציפית ההגדרות של User נשמרות תחת הנת"ב הבא:

```
%programdata%\Microsoft\GroupPolicy\Users
```

הסיבה שזה מעניין אותנו היא שהתקיימה %programdata% ניתנת לכתיבה באופן דיפלוטי על ידי משתמשים חלשים.

עוד משהו שקורה בזמן תהליך העדכון, הוא שהתהליך GPClient מבצע פעולה SetSecurityFile - בקיצור, מאפשרת כתיבה ל-DAACL של הקובץ, הוא מבצע את הפעולה בהרשאות שלו, שהן System:



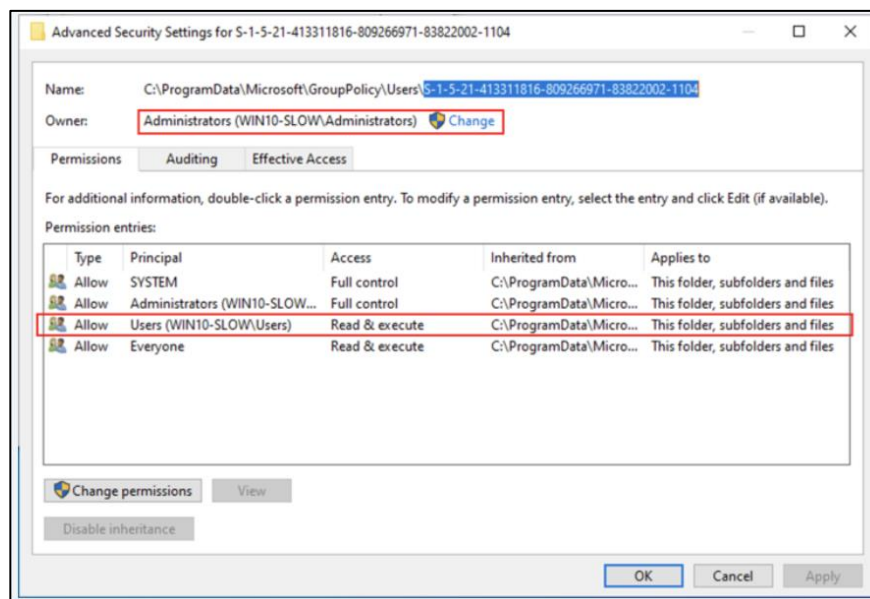


המטרות הסופיות שלנו יהיה:

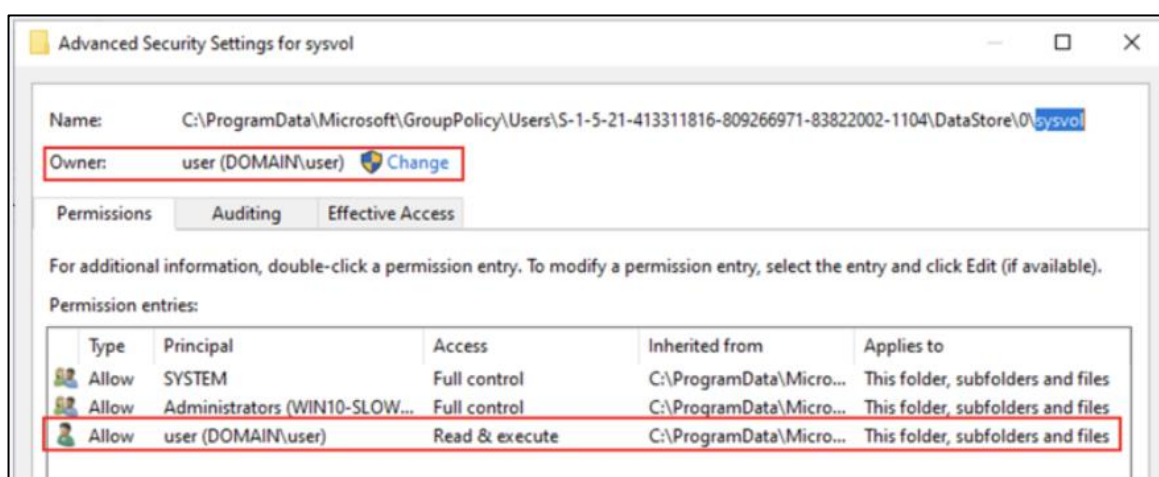
1. למצוא השמת הרשאות מתירנית שמתבצעת על ידי השירות הזה
2. ליצור קובץ בתקיה שמצביע (link, junction) לקבצים שנרצה להשיג הרשאות כתיבה עליהם
3. לגרום לשירות לתת לנו הרשאות על הקובץ link

היררכיית התקיות כרגע היא כזו:

בתוך ה-Group Policy יש תקיה עם ה-SID של המשתמש, על תקיה זו אין למשתמש הרשאות write, רק Read Execute



אבל- אם יורדים מטה בהיררכיית התקיות, נראה שיש תקייה בשם sysvol שהמשתמש הוא ה-Owner שלה:





אז בעצם המשתמש הזה יכול לבטל ירושה על התקיייה הזו ולהעניק לעצמו הרשאות מסוג Full Control על כלל הקבצים בתקיייה. לאחר מכן, כשתתחיל פעולת GPUUpdate (שרצה בהרשאות SYSTEM) תורץ הפקודה SetSecurityFile וישוכתב ה-DAACL של כלל הקבצים תחת התקיה הזו.

אז כעת, נוכל ליצור קובץ מסוג לינק (או junction) מתוך התקיה הזו (שכן אנחנו ה-Owners שלה) אל מיקום אחר שאין לנו הרשאות כתיבה אליו ובכך לגרום ל-Service לבצע SetSecurityFile על המיקום הזה!

נראה שכאשר אנחנו מפנים את הקובץ אל מיקום אחר, ה-Service אכן פונה אליו ומבצע את SetSecurityFile עליו ועל כל ה-SubFolders שלו. אך ההרשאות שה-Service מחיל לא מאפשרות לנו גישה מלאה לקבצים, אלא רק Read Execute.

בתהליך המחקר, התגלה כי אם במהלך ביצוע הפעולה SetSecurityFile ה-Service נכשל (כי אין לו הרשאות או כי אין לו יכולת לפתוח את הקובץ כי הוא נעול כבר על ידי קובץ אחר), הוא מעניק הרשאות Full Control ל-User שלנו!

מימוש התהליך יהיה כזה:

1. נכנס לתקיה sysvol המקומית בה יש לנו הרשאות Owner
2. נבטל ירושה ונעניק לעצמינו Full Control על התקיייה
3. ניצור קובץ שהוא בעצם Junction לתקיייה אחרת שאין לנו גישת Write אליה (למשל, C:\Program (Files)\VMware
4. ניצור תקיה נוספת עם קובץ שנגדיר עליו OpLock (מה שימנע מה-Service לבצע Write DACL) כדי ליצור את השגיאה
5. נריץ gpupdate, מה שיגרום ל-GPClient לרוץ
6. GPClient יבצע את הפקודה SetSecurityFile עד שיגיע לקובץ הנעול ושם יקבל שגיאה
7. ברגע שמתקבלת השגיאה, נמחק את ה-Junction ונשחרר את ה-OpLock
8. משום שה-Write DACL עבד רק באופן חלקי וה-Junction נמחק, נקבל על התקיה הרשאות Full Control
9. כעת נוכל לשנות את תוכן התקיה ולעשות בה כרצוננו ☺

והינה קישור לעמוד GitHub שמכיל כלי קטן שממש את המתקפה הזו:

<https://github.com/thezdi/PoC/tree/master/CVE-2020-16939>

המטרה של ההסבר הזה היתה לפתוח אתכם לעולם של חולשות GPO ולהבין את הקונספט באופן כללי, מעבר לזה אצרף קישורים בסוף המאמר לחולשות דומות.



בואו נדבר הגנה

Group3r

למי שיש לו את היכולת להוריד כלי מהאינטרנט ולהריץ בסביבה, יש אחלה כלי בשם Group3r שמוציא לכם המון אינפורמציה על הסביבה שלכם בצורת דוח קצת מבולגן אבל מאוד אינפורמטיבי. קישור לכלי:

<https://github.com/Group3r/Group3r/tree/main>

דוגמא לחלק מהפלט - הנה ה-logon script שלנו:

```
2023-07-05 13:27:20 +00:00 [GPO]
| GPO | MyFirstGPO {331A6179-32FC-4A0D-B808-8112990E912A} Current |
|-----|-----|
| Date Created | 7/5/2023 9:07:23 AM |
| Date Modified | 7/5/2023 10:03:45 AM |
| Path in SYSVOL | \\d01.lab\sysvol\d01.lab\Policies\{331A6179-32FC-4A0D-B808-8112990E912A} |
| Computer Policy | Enabled |
| User Policy | Enabled |
| Link | DC=d01,DC=lab (Enabled, Unenforced) |
|
| Setting - User Policy | Script |
|-----|-----|
| Script Type | Logon |
| CmdLine | new.bat |
|
| Finding | Black |
|-----|-----|
| Reason | Writable Logon script file identified at |
| | \\d01.lab\sysvol\d01.lab\Policies\{331A6179-32FC-4A0D-B808- |
| | 8112990E912A}\User\Scripts\Logon\new.bat |
| Detail | This script will run in the context of the users/computers to which this GPO is |
| | applied. Change the script, get command exec as those users/computers. |
```

חשוב לציין שהכלי יוציא כפלט את כלל ההגדרות שלכם, מה שעלול ליצור דוח מאוד מבולגן וגדול במיוחד, יש פרמטר שמוציא רק את ה-Findings.

לעצלנים מבינינו - כתבתי כלי שמפרסר את הלוג של Group3r ומוציא GUI נחמד שיותר כיף לחטט בו. קישור לגיטהאב של הכלי:

<https://github.com/sap8899/Group3rExplorer>



ההמלצה שלי- תממשו!

אז הנה כמה סקריפטים נחמדים שיוכלו להכניס אתכם לעניינים, הסקריפט הזה מבוסס על WindowsAPI והוא יפרסר לכם קבצי inf/.ini.

כל מה שאתם צריכים להכניס הוא את ההגדרות הבאות:

- הנתיב לקובץ
- הסוג הגדרות שאתם מחפשים (App) - למשל Privilege Rights
- והקונפיגורציה הספציפית (key) - למשל הרשאות מסוג SeTcbPrivilege

```
using System;
using System.Runtime.InteropServices;
using System.Text;
using System.Threading.Tasks;
namespace GPPS
{
    public class Class1
    {
        [DllImport("kernel32.dll", CharSet = CharSet.Unicode)]
        public static extern uint GetPrivateProfileString(
            string lpAppName,
            string lpKeyName,
            string lpDefault,
            StringBuilder lpReturnedString,
            uint nSize,
            string lpFileName);
        public static string[] getValue(string fullPath, string app, string key)
        {
            StringBuilder sb2 = new StringBuilder(500);
            uint res2 = GetPrivateProfileString(app, key, "", sb2, (uint)sb2.Capacity,
            fullPath);
            string[] resultsArray = new string[2];
        }
    }
}
```

```
if(res2 > 0)
{
    resultsArray[0] = fullPath;
    resultsArray[1] = sb2.ToString();
}
else
{
    resultsArray[0] = fullPath;
    resultsArray[1] = "-1";
}

return resultsArray;
}

public static Task<string[]> Check(string fullPath, string app, string key)
{
    return Task.Run(() => getValue(fullPath, app, key));
}

}

class Program
{
    static void Main(string[] args)
    {
        string fullPath = "Path";
        string app = "App";
        string key = "Key";

        var resultTask = Class1.Check(fullPath, app, key);
        resultTask.ContinueWith(task =>
        {
            string[] results = task.Result;
            string filePath = results[0];
            string value = results[1];

            Console.WriteLine("File Path: " + filePath);
            Console.WriteLine("Value: " + value);
        });

        Console.ReadLine();
    }
}
}
```

בואו נראה הרצה לדוגמא, שניתי את הפרמטרים לפרמטרים הבאים:

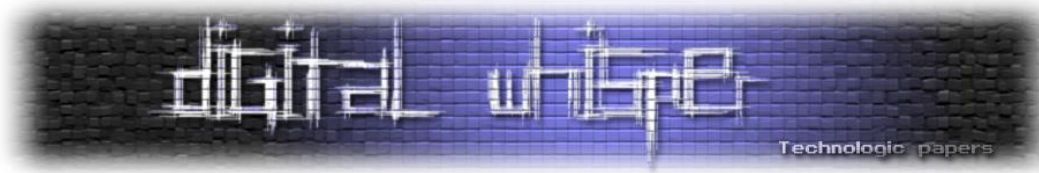
```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        string fullPath = "C:\\Temp\\GPO\\GptTmpl.inf";
        string app = "Privilege Rights";
        string key = "SeTcbPrivilege";
    }
}
```

והנה הפלט:

```
C:\security_research\ideas\gpo\iniParser\iniParser\bin\Debug\net6.0\iniParser.exe
File Path: C:\Temp\GPO\GptTmpl.inf
Value: normal1
```

באותה מידה נוכל להשתמש בקוד כדי לחפש נתיבי Registry של הגדרות ספציפיות כמו Ldap Signing.

1. קוד PS שקורא את כל המדיניות בדומיין ומציג לכל Container אילו מדיניות מוחלות עליו:



```

$ldapFilter = "(&(|(objectClass=organizationalUnit)(objectClass=site)(objectClass=domain))(gplink=*))"
$attributesToRetrieve = @("gplink")
$ldapSearcher = New-Object System.DirectoryServices.DirectorySearcher
$ldapSearcher.Filter = $ldapFilter
$ldapSearcher.PropertiesToLoad.AddRange($attributesToRetrieve)
$searchResults = $ldapSearcher.FindAll()

foreach ($result in $searchResults) {
    $entry = $result.GetDirectoryEntry()
    $gplinks = $entry.Properties["gplink"].Value
    $gpList = $gplinks.split("\").split("\")
    Write-Host "Container: $($entry.distinguishedName)"
    foreach ($link in $gpList)
    {
        $temp = $link -match "LDAP://cn=((([A-Z0-9a-z]+){4}[A-Za-z0-9]+)), "
        if($temp -and $Matches[1])
        {
            write-host "Policy: $($Matches[1])"
        }
    }
    Write-Host "-----"
}
$searchResults.Dispose()

```

דוגמא לפלט:

```

Container: OU=Domain Controllers,DC=d01,DC=lab
Policy: {2E2F2159-D01D-4C76-A147-F2ACBF7766BD}
Policy: {6AC1786C-016F-11D2-945F-00C04FB984F9}
-----
Container: OU=TEST,DC=d01,DC=lab
Policy: {a6622496-db50-47e8-a704-b8ebfd5ccac1}
Policy: {17596372-1f24-4d86-b4b6-eee15cf9f354}
Policy: {4b76de20-2b2d-463e-b669-0699f627561c}
Policy: {15e2fc92-49e9-4018-80e4-6b9327026ec5}
Policy: {a1b58d97-3559-4a57-a2bb-eea6027a4296}
Policy: {5a1612b0-01bf-4980-9fb5-b27f6348c410}
Policy: {6d6586c4-79fa-481c-b81e-7048ba847f1d}
Policy: {2899ceac-8d90-45bc-81d4-f9641da2b81a}
Policy: {1dc7000d-9f41-4b49-a1d9-728f62a93771}
Policy: {2ada352c-ea0d-455d-a6cd-432f3d2b6ad3}
Policy: {220683bf-9054-4ae3-844d-d7d4726898a8}
Policy: {c9db5fa6-0525-4688-a027-0bcd13725442}
Policy: {9444bf5e-7d69-42b2-bb56-eadb1861a0f}
Policy: {ef0cf8ea-00b9-4a55-b936-74abc38c8940}

```

2. סקריפט שעובר על כל ה-GPO בדומיין ומוציא את הנתבי שלהם, ה-GUID של המדיניות, השם של

המדיניות (במידה וקיים) ודגלים (האם Enabled\Disabled):

```

$ldapFilter = "(&(objectClass=groupPolicyContainer))"
$attributesToRetrieve = @("gpcfilesyspath", "flags", "cn", "displayName")
$ldapSearcher = New-Object System.DirectoryServices.DirectorySearcher
$ldapSearcher.Filter = $ldapFilter
$ldapSearcher.PropertiesToLoad.AddRange($attributesToRetrieve)
$searchResults = $ldapSearcher.FindAll()

foreach ($result in $searchResults) {
    $entry = $result.GetDirectoryEntry()
    $flags = $entry.Properties["flags"].Value
    $status = switch ($flags)
    {
        "0" {"User policy enabled. Computer policy enabled."}
        "1" {"User policy disabled. Computer policy enabled."}
        "2" {"User policy enabled. Computer policy disabled."}
    }
}

```



```
"3" {"User policy disabled. Computer policy disabled."}
}
Write-Host "PolicyStatus: $($status)"
foreach ($attribute in $attributesToRetrieve) {
    $value = $entry.Properties[$attribute].Value
    Write-Host "$($attribute): $($value)"
}
Write-Host "-----"
}
$searchResults.Dispose()
```

דוגמא לפלט:

```
PolicyStatus: User policy enabled. Computer policy enabled.
gpclistpath: \\d01.lab\sysvol\d01.lab\Policies\{0B18E105-1B7E-4F1F-9547-07A7D50D81E5}
flags: 0
cn: {0B18E105-1B7E-4F1F-9547-07A7D50D81E5}
displayName: TestGPO254
-----
PolicyStatus: User policy enabled. Computer policy enabled.
gpclistpath: \\d01.lab\sysvol\d01.lab\Policies\{CE9939E8-2794-4C69-9DEE-6C0A7C039795}
flags: 0
cn: {CE9939E8-2794-4C69-9DEE-6C0A7C039795}
displayName: TestGPO255
-----
```

נוכל לקרוא לכל ה-Child object בנתיב של המדיניות מסוג ini/inf ולפרסר אותם באמצעות הסקריפט CPP שלנו! (הידעתם - אפשר לכתוב CPP בתוך IPS)

לאחר הפרסור, נוכל לחפש כל הגדרה שמעניינת אותנו, ולוודא שאין הגדרות שמסכנות את הסביבה שלנו.

- לא הוספתי כאן סקריפט שמפרסר XML למרות שיש מדיניות שמוגדרות גם בתוך קבצי מסוג זה, משום שממש קל למצוא בכל מקום באינטרנט היום קוד בכל שפה אפשרית לפרסור XML ©

רעיונות לזיהוי מיסקונפיגורציות ב-GPO שיכולות להוביל להסלמת הרשאות, הרצת קוד מרוחק וכו'

- הרשאות על סקריפטים שנמצאים בתקיה שניתנת לכתיבה על ידי משתמשים חלשים. (למשל, תקייה שיתופית שנמצא עליה סקריפט שרץ בכל logon של משתמש). אם משתמש חלש יכול לשנות את הסקריפט, הוא יכול לשים שם סקריפט זדוני כרצונו.
- אותו הקונספט חל על הרשאות על משימות מתוזמנות
- מדיניות על OU-ים ספציפיים שמאפשרות הגדרות דומייניות שונות כמו למשל:
 - Cached Credentials
 - LM Hash
 - LDAP Signing
- לא נגעתי בנושא הזה הרבה אבל ניתן לבצע גם פעולות על קבצים דרך מדיניות GPO וגם יש כמה CVE מעניינים, מצורף בסוף המאמר
- התנהגות ה-GPClient שכנראה רץ לחלקינו על העמדות ושווה להכיר



סיכום

בתור מגנים, וגם תוקפים. חשוב להכיר את הסביבה שאתם עובדים עליה, וחלק מכובד מהסביבה הוא ה-GPO. כל כך הרבה הגדרות, משחקי הרשאות והרצת קוד מרוחק מתאפשרים באמצעות הגדרות ה-GPO, לכן חשוב לוודא שכל ההגדרות מקונפגות כפי שצריך, ושאר תוקף לא ניצל את ה-GPO כדי להשאיר Backdoor, להסלים הרשאות או להריץ קוד על הדומיין שלכם. מקווה שנהנתם!

על המחברת

@sapirxfed - עושה retweet לכל דבר שקשור ל-AD או AAD ☺
אוהבת לכתוב קוד, מנסה למצוא חולשות, ומעריצה שרופה של Digital Whisper!

תודות

תודה ענקית לקולגה שלי **Andrea Pierini (@decoder_it)** שלא יבין מילה מהמאמר הזה אבל הקרדיט הזה בהחלט מגיע לו.

ביבליוגרפיה

כל הדוקומנטציה של מיקרוסופט על GPO:

- <https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/user-rights-assignment>
- <https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/applying-group-policy>
- <https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/overriding-and-blocking-group-policy>
- <https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/group-policy-hierarchy>
- <https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/linking-gpos-to-active-directory-containers>
- <https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/group-policy-objects>

קישורים למספר מאמרים מעניינים של אנדריאה בנושא GPO:

- <https://decoder.cloud/2022/04/25/a-not-so-common-and-stupid-privilege-escalation/>
- <https://decoder.cloud/2022/04/27/group-policy-folder-redirection-cve-2021-26887/>



- <https://decoder.cloud/2023/02/16/eop-via-arbitrary-file-write-overwrite-in-group-policy-client-gpsvc-cve-2022-37955/>

עוד קצת CVE:

- <https://www.zerodayinitiative.com/blog/2020/10/27/cve-2020-16939-windows-group-policy-dacl-overwrite-privilege-escalation>
- <https://www.cyberark.com/resources/threat-research-blog/group-policies-going-rogue>

הרשאות משתמש מעניינות:

| משמעות | שם ההרשאה |
|--|---------------------------------|
| מאפשר להוסיף לעצמך הרשאות Owner על קבצים, מפתחות Registry וכו' | SeTakeOwnershipPrivilege |
| מאפשר לדרוס או לשנות כל קובץ | SeRestorePrivilege |
| משתמש בעל הרשאות אלו, יכול ליצור אפליקציה שקורית ל-Credential Manager שיחזור להחזיר פרטי הזדהות של משתמש אחר | SeTrustedCredManAccessPrivilege |
| מאפשר התחזות למשתמשים אחרים | SeTcbPrivilege |
| מאפשר התחזות, משומש הרבה במתקפות ה-Potato | SeAssignPrimaryTokenPrivilege |
| מאפשר לדרוס הרשאות על קבצים ותיקיות ובכך לקרוא כל קובץ במערכת הקבצים | SeBackupPrivilege |
| מאפשר ליצור לעצמך כל גישה שתרצה | SeCreateTokenPrivilege |
| כל מה שקשור ל-Mimikatz: dump לתהליכים, להשיג Handle לתהליכים וכו' | SeDebugPrivilege |
| מאפשר להתחזות למשתמשים אחרים | SeImpersonatePrivilege |
| מאפשר לטעון דרייברים | SeLoadDriverPrivilege |
| משתמש בעל הרשאות אלו, יכול ליצור אפליקציה שקורית ל-Credential Manager שיחזור להחזיר פרטי הזדהות של משתמש אחר | SeTrustedCredManAccessPrivilege |
| מאפשר התחזות למשתמשים אחרים | SeTcbPrivilege |
| מאפשר לדרוס או לשנות כל קובץ | SeRestorePrivilege |