**המבחן של תומר מור**

1. **נתחיל בבסיס – מה זה Domain? אבל תפרט.**

\*\*\*אני רשמתי על דומיין של אתר אינטרנט, ואז הבנתי שכנראה התכוונת ל network domain אבל בגלל שכבר רשמתי, אני אשאיר את הwebsite domain ורק אוסיף על network domain

Network domain

אוסף של התקנים ומשאבים המחוברים לרשת, המנוהלים כיחידה אחת תחת בקרה ניהולית משותפת.

זה למעשה פועל כקבוצה אדמינסטרטיבית של מחשבים ורכיבי רשת אחרים, תוך שיתוף מדיניות אבטחה משותפת ומידע על חשבון משתמש.

דומיין מאפשר ניהול מרכזי של משאבי רשת. משמעות הדבר היא שמנהלי מערכת יכולים לשלוט בגישת משתמשים, הגדרות אבטחה ומדיניות רשת אחרת ממיקום מרכזי.

אפשר לנהל את הדומיין דרך אינטרפייס שמייקרוסופט יצרו וגם דרך CLI (Command line interface).

כל המחשבים בתוך דומיין חולקים מדיניות אבטחה משותפת ומידע על חשבון משתמש. זה מפשט את אימות המשתמש ובקרת הגישה, ומקל על ניהול האבטחה בתוך הרשת.

חשוב לציין שדומיין אינו בהכרח מרחב פיזי, זהו מושג אדמיניסטרטיבי שיכול להשתרע על פני מספר מקומות פיזיים.

Website domain

**בקצרה:** כתובת קריאה על ידי בני אדם עבור אתר אינטרנט באינטרנט, המאפשרת גישה קלה יותר לאתרים מאשר שימוש בכתובות IP מספריות, לדוגמא כאשר אני רושם <https://www.google.com> הנתיב הזה מומר מאוחרי הקלעים ל <https://142.250.75.78>, דרך אחת לבדוק דברים כאלה זה בעזרת פקודה פשוטה אשר נקראת ping

A black screen with white text

AI-generated content may be incorrect.

**איך זה עובד?**

1. **Ip addresses and DNS**

* כל אתר מתארח על שרת עם ip מספרי לדוגמא: 192.0.2.172
* הDNS (Domain Name System) פועל כמו ספר טלפונים ומתרגם את שמות הדומיין לכתובת IP המתאימות להם. (הDNS הראשי של גוגל הוא 8.8.8.8 והמשני הוא 8.8.4.4)
* כאשר מקלידים את הדומיין בדפדפן הDNS מחפש את הכתובת IP ומנתב את הדפדפן לשרת הנכון

1. **מבנה שם הדומיין**

* שם דומיין מורכב מכמה חלקים אשר קוראים אותם מימין לשמאל.
* הצד הכי ימני הוא הTLD (Top Level Domain) כמו .com, המציין את סוג או מטרת האתר.
* שאר החלקים (Subdomains) מפרטים עוד יותר את מבנה האתר

1. **יתרונות השימוש בשמות דומיין**

* זכירות – קל יותר לזכור מאשר ip מספרי
* מיתוג – מאפשר ליצור זהות מותג מוכרת
* נגישות – מאפשר למשתמשים למצוא ולגשת לאתר בקלות
* גמישות – ניתן לשנות את השרת המאחסן את האתר מבלי לשנות את שם הדומיין (רק משנים את המיפוי)

1. **מה השוני בין דומיין לבין Workgroup?**

Workgroup זה רשת עמית לעמית (peer-to-peer) שבה כל המחשבים זהים וכל אחד מהם מנהל את האבטחה והמשאבים שלו. דומיין, לעומת זאת, היא רשת מרכזית המנוהלת על ידי שרת אחד או יותר (domain controller) המספקים אבטחה מרכזית וניהול משתמשים.

הבדלים:

* **מרוכז** – דומיין הוא מערכת ניהול מרוכזת וworkgroup היא מבוזרת (Decentralized), זה אומר שכל ההגדרות של מחשבים, אבטחה וכו, מנוהלים במקום אחד (דומיין) או בכל מחשב בפני עצמו (workgroup)
* **הרחבה** – דומיין ניתן להרחיב בקלות מכיוון שאין בו תלות במי שקשור אליו או כמה (כמובן שאם יש יותר מידי צריך להרחיב) ולעומת זאת workgroup מוגבל עד כדי מכיוון שכל מחשב אחראי על עצמו, וזה ממש קשה לשמור על מחשב אחד (לדוגמא עם גרסאות) כמו שצריך, עכשיו כמה? אוף
* **אבטחה** – דומיין בטוח יותר מכיוון שכל הניהול נמצא במקום אחד ולא מפוזר בכמה מחשבים.

כאשר עובדים עם רשת גדולה עדיף להשתמש בדומיין, לא רק שהניהול שלא הרבה יותר קל, הוא מותאם לכמויות גדולות של משתמשים בו זמנית וworkgroup הוא יותר כמו שהוא נשמע... קבוצת למידה של בית ספר של עד 10 אנשים ככה.

דומיין יכול לנהל ולשלוט על משאבי הרשת במקום אחד מרוכז, מה שworkgroup לא יכול.

1. **מהו Forest, וכיצד באופן הכי כללי הוא מתקשר ל Domain?**

AD Forest (Active Directory Forest) הוא המכולה הלוגית ביותר בAD המכילה תחומים (domains), משתמשים, מחשבים ומדיניות קבוצתית.

AD אחד יכול להכיל יותר מDOMAIN אחד, ואנחנו קוראים לשכבה שמעל הדומיין AD Forest

מתחת לכל דומיין יכול להיות כמה עצים.

**עץ** – אוסף של אחד או יותר דומיינים שחולקים את אותו מרחב שמות (namespace) וסכימה משותפת.

בגדול ההירכיה היא ככה:

יער

עץ

דומיין

שכבה עליונה נוספת זו יוצרת אתגרי אבטחה ופוטנציאל מוגבר לניצול, אך היא יכולה גם להוביל לבידוד ואוטונומיה גדולים יותר בעת הצורך.

**איך domains מתקשרים בתוך הforest** (אולי לא הבנתי את השאלה, ניחה)

בתוך הForest הdomains מתקשרים בעזרת יחסי אמון דו-כיווניים, טרנזיטיביים, אשר נוצרים אוטומטית בין דומיינים באותו הforest

היחסים בין יערות לא חייבים להיות דו כיוונים, הם יכולים להיות חד כיוונים (יער אחד יכול לגשת למשאבים של יער אחר, אבל היער האחר לא יכול).

הקבוצה החזקה ביותר היא: Enterprise Admin (EA) group

1. **מהו תפקידו של הDC?**

עשיתי במבחן בית.

במבחן בית רשמת:

DC- שרת אחראי על ניהול בקשות אבטחת רשת וזהות. הוא פועל כשומר סף ומאמת האם משתמש מורשת לגשת למשאבים בדומיין.

\*\*\* תוסיף עוד הסבר על DC, איך הוא מבצע את ניהול האבטחה, מה יש עליו, מי יכול להתחבר אליו, האם יש DC 1 ? אחרי שהוא אימת משתמש, איך הוא יודע איזה הרשאות יש למשתמש? וכו.....

[לעשות!!!]

1. **איפה מאוחסן הAD? הסבר**

הAD Database מאוחסן בעיקר בנתיב **%systemroot%\NTDS** בתוך קובץ אשר נקרא **NTDS.dit** על domain controller (שרת).

* NTDS.dit – זה קובץ מסד הנתונים הראשי אשר מחזיק את כל המידע של הAD, כולל מידע על משתמשים, קבוצות, מחשבים ואובייקטים אחרים הקשורים לAD

הקבוץ בנוי בעזרת Jet Blue (ESE) database format לקובת יש את הרכיבים הבאים:

* **טבלת נתונים** – כל הנתונים הקריטיים של מסד הנתונים, אובייקטים, attributes וכו'.
* **טבלת קישורים** – קישורים בין אובייקטים בAD
* **טבלת SD** – Security descriptor (ACLs)
* **טבלת סכימה** – מתאר סוגי ותכונות אובייקטים
* **טבלת metadata** – עוקב אחר שינויים, כמו רפליקציות.

הקובץ בנוי ככה בשביל:

* גישה מהירה
* מעקב אחרי העתקים
* אחסון מאובטח של credentials

אי אפשר לגשת לקובץ הזה סתם ככה מכיוון:

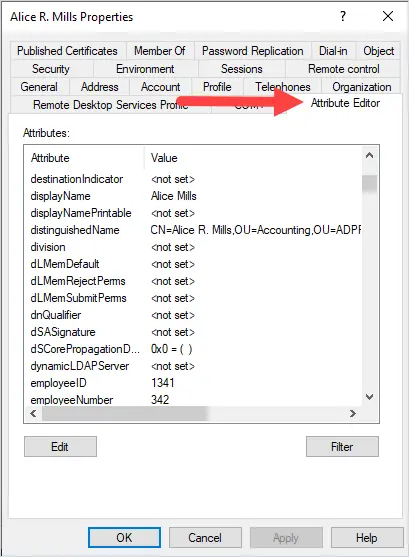
* כדי למנוע שחיתות או התעסקות הוא נעול בזמן ריצה
* הקובץ משומש על ידי התהליך LSASS ולכן העתקה רגילה לא תעבוד מכיוון שהוא נעול בצורה אקסקלוסיבית מווידוס
* חייב הרשאות admin או SYSTEM כדי לקרוא או לגשת לקובץ, וגם אז, הפקודות copy או xcopy לא יעבדו בגלל נעילות שפועלות.

1. **מהם Attributes? איפה ניתן למצוא Attributes ב Active Directory? האם למשתמש יש Attributes?**
2. מהם Attributes?

פיסות מידע או נתונים המגדירים את המאפיינים של אובייקטים המאוחסנים במסד הנתונים של הAD

1. איפה ניתן למצוא Attributes ב Active Directory?

ניתן למצוא אותם בתוך חלון המאפיינים של אובייקט (כגון משתמש, קבוצה או מחשב), לאחר הפעלת התכונות המתקדמות, הטאב המציג Attributes נקרא "Attribute editor"



1. האם למשתמש יש Attributes?

כן? לכל אובייקט בActive directory יש Attributes ומכיוון שמשתמש זה אובייקט, אז כן.

Attribute יכול להיות כל מה שתרצי, יש אפילו בהמשך על יצירת attribute אבל ארשום כאן כמה מהדברים שבאים built in:

* sAMAccountName – שם logon (0pre-Windows 200)
* distinguishedName – הדרך המלאה לאובייקט בLDAP
* mebmerOf – רשימה של הקבוצות אליה האובייקט משוייך

1. **מה זה SSP?**

Service Switching Point, מתג תקשורות, הממוקם לרוב במשרד המרכזי, המטפל בשיחות באמצעות אינטראקציה עם נקודת בקרת שירות (SCP).

הSSP משתמש בפרוטוקולי SS7 (Signaling System 7), ספציפית בIntelligent Network Application Part (INAP), לשלוח שאילתות לSCP עבור מידע ניתוב ונתונים אחרים הקשורים לשירות.

בעיקרו, הSSP משמש כשער (gateway) לרשת חכמה, ומאפשר תכונות מעבר לניתוב שיחות בסיסי.

**SSP (Security Support Provider)**

הוא קובץ DLL שמספק שירותים לאבטחה דרך ממשק שנקרא SSPI (Security Support Provider Interface) . מדובר בשכבת תקשורת בין האפליקציה לבין מנגנוני האבטחה של Windows כמו NTLM, Kerberos ועוד.

על מה הוא אחראי:

* אימות הזדהות (Authentication)
* הצפנה ופרוטוקולים מאובטחים (SSL/TLS)
* שמירה על שלמות ההודעה (integrity)
* תמכיה באימות הדדי (Mutual authentication)

דוגמאות לSSP נפוצים:

* **NTLM** – msv1\_0.dll
* **Kerberos** – Kerberos.dll
* **Schannel** – schannel.dll – תעבורה מוצפנת
* **Negotiate** – negotiat.dll או secur32.dll – בוחר אוטומטית בין NTLM או Kerberos
* **CredSSP** – credssp.dll – משמש ל-RDP

מתי SSP נכנס לתמונה?

* בזמן התחברות לwindows
* בעת שימוש ב-RDP
* בזמן גלישה באינטרנט באתרים מאובטחים (https)
* כשאפליקציות מבצעו. אימוץ מול השרת (למשל IIS)

1. **הסבר על msv1\_0**

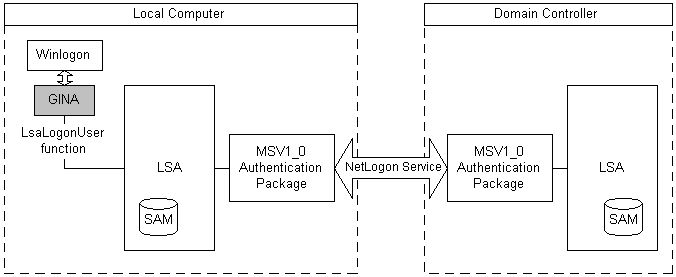
LSA (Local security Authority) – תת מערכת המאמתת ומכניסה משתמשים למערכת מקומית, LSA גם מתחזקת מידע על כל היבטי האבטחה המקומית במערכת.

GINA (Graphical Identification and Authentication)- זה DLL אשר ניתן להחלפה שנטען על ידי הwinlogon, GINA מיישמת את מדיניות האימות של מודל הכניסה האינטראקטיבי וצפוי לבצע את כל אינטראקציות הזיהוי והאימות של המשתמש.

הLSA קורא לmsv1\_\_0 כדי לעבד את המידע של ההתחברות הנלקח מGINA בשביל הwinlogon logon process.

דוגמא לחבילת אימות של ווינדוס, החבילה מקבלת שם משתמש וhashed password, אותם היא מחפשת במסד הנתונים של מנהל חשבונות האבטחה (SAM, יש תיאור שלו בתרגיל בית).

Msv1\_\_0 תומך גם בכניסה דומיינית (domain logon). הdomain logon של חבילה זו משתמש באימות מעבר (pass-through authentication) כנראה למטה.



באימות מעבר (pass through authentication) – המופע המקומי של msv1\_\_0 משתמש בשירות netlogon כדי לקרוא למופע של msv1\_\_0 בפועל בבקר התחום (domain controller).

לאחר מכן, המופע של msv1\_0 בבקר התחום (domain controller) בודק את מסד הנתונים SAM של בקר התחום (domain controller) ומחזיר את תוצאות הכניסה למופע של msv1\_0 במחשב במקומי. הגרסה המקומית של msv1\_0 מעבירה את תוצאות הכניסה למופה של הLSA במחשב המקומי.

אם בקר התחום (domain controller) אינו זמין, וה-LSA מכיל אישורים המאוחסנים במטמון עבור המשתמש, המופע המקומי של msv1\_0 יכול לאמת את המשתמש באמצעות נתוני הכניסה המאוחסנים במטמון (אני מניח שזה גם למה אפשר להכנס עם חוגר שהתחברתי אליו קודם למחשב גם כשאין רשת?)

1. **הסבר על wdigest**

זה פרוטוקול אימות אשר משומם במערכות ההפעלה של ווינדוס. הוא הוצג בווינדוס XP

והוא מאופשר כברירת מחדל (default). הוא עוצב כדי לשפר את הנוחות של אימות אפליקציות HTTP וSASL בכך שהוא שומר במטמון (cache) את הסיסמא הלא מוצפנת (clear text)

זה מאפשר למשתמשים להתחבר לשירותים מבלי להכניס את הcredentials כל הזמן.

אבל, הגישה הזו שלWDIGEST של שמירת הסיסמאות הלא מוצפנות בזיכרון יוצר פרצת אבטחה אשר תוקפים יכולים לנצל בעזרת כלים כמו Mimikatz.

אחד מהSSP הוותיקים של windows שמספק אימות מבוסס Digest כלמור שיטת אימות שבה הסיסמא לא נשלחת כטקסט גלוי, אלא בצורה מעובדת. השיטה הזו שימשה בעבר בעיקר באימותי HTTP ו LDAP

שם הdll: wdigest.dll

**מטמון**

זהו מנגנון אשר מאפשר לשמור עותקים של משאבים (כמו קבצי CSS, HTML, תמונות, JSON וכו) באופן מקומי או בשרתים מתווכים – כדי להאיץ טעינה ולחסוך בתעבורת רשת.

איך זה עובד?

כאשר הדפדפן מקבל תגובה מהשרת, הוא מסתכל על כותרות (headers) שקובעות אם מותר לו לשמור את הקובץ וכמה זמן, לדוגמא:

Cache-Control: max-age=86400

זה אומר לדפדפן – "תשמור את הקובץ הזה למשך 24 שעות, אין צורך לבקש שוב מהשרת"

בטעינה הבאה הדפדפן יטען את הקובץ מהזיכרון המקומי במקום לבקש שוב

בHTTP המטמון יכול להישמר בכמה רמות שונות, בהתאם לסוג המטמון ומי מבצע אותו – דפדפן, שרת,פרוקסי, או שרת קצה.

סוגים:

1. **מטמון צד לקוח (דפדפן)** – שומר עותקים של בקשות/תגובות, כדי לטעון משאבים מהר יותר בלי לשלוח שוב בקשה לשרת, זה נשמר פיזית על המחשב של המשתמש, בכרום: C:\Users\<USERNAME>\AppData\Local\Google\Chrome\User Data\Default\Cache  
   בפיירפוקס**:** C:\Users\<USERNAME>\AppData\Local\Mozilla\Firefox\Profiles\<PROFILE>\cache2
2. **מטמון צד שרת / פרוקסי / CDN (מטמון ביניים)** – מערכות כמו NGINX, Cloudflare או Squid יכולות לבצע caching של תגובות כדי להקל על שרת המקור.  
   נשמר בשרת עצמו או במיקום מרכזי (shared cache)  
   לדוגמה CDN שומר את הקובץ קרוב גיאוגרפית למשתמש, NGINX ןSQUID שומרים קבצים בדיסק או בזיכרון
3. **מטמון בזיכרון RAM** – חלק מהדפדפנים או פרוקסי שומרים קבצים באופן זמני בRAM, ביצרון של זה הוא זה מהיר יותר אך החיסרון, נעלם אחרי ריסטרט

איך HTTP יודע אם לשמור במטמון?

בעזרת Http headers

* **Cache-Control** – הכללים של המטמון (max-age=3600,no-cache וכו)
* **Expires** – מתי פג התוקף של המשאב
* **ETag** – מזהה ייחודי של התוכן – אם לא השתנה, אין צורך להוריד שוב
* **Last-Modified** – תאריך עדכון אחרון של הקובץ
* **Vary** – קובע לפי אילו תנאים מותר להשתמש במטמון (למשל לפי Accept-Encoding)

איך עובד תהליך ההתאמתות עם wdigest?

1. השרת שולח אתגר (challange) – שרת (למשל שרת LDAP) שולח בקשה ללקוח להזדהות, לדוגמא:  
   WWW-Authenticate: Digest realm="example.com", nonce="abc123", qop="auth"
2. הלקוח מגיב עם Digest Response – הלקוח (למשל מחשב Windows שמנסה להתחבר) מחולל hash מבוסס על:
   1. שם משתמש
   2. סיסמא
   3. Realm (תחום)
   4. Nonce (מספר חד פעמי)
   5. URI
   6. סוג הפעולה (GET/POST)

דוגמא: Authorization: Digest username="user", realm="example.com", nonce="abc123",

uri="/login", response="4a7d1ed414474e4033ac29ccb8653d9b"

1. השרת בודק אם הhash מתאים – השרת יוצר בעצמו את אותו hash מתוך בסיס הנתונים שלו, אם הערכים זהים, המשתמש מאומת.

\*\* עכשיו כשאני חושב על זה ואני רואה את הheaders עשיתי גם משהו שקשור לזה, יצרתי שרת IIS שמתחבר לLDAP ולוקח את המספר האישי של משתמש על פי החוגר (יותר נכון על פי המשתמש המחובר) וראיתי את הheaders האלה כשעשיתי debug לזה, מגניב לדעת מאיפה זה מגיע.

1. **הסבר על NTLM ולאילו התקפות פגיע פרוטוקול NTLM? הסבר על כל אחד מהמונחים הבאים: Net-NTLMv2, Net-NTLMv1, LM HASH , NTLM HASH**

הסברתי על NTLM במבחן בית.

הערות - אצטרף תמונות

איזה התקפות פגיע?

ב NTLM (Net-NTLMv1 וגם Net-NTLMv2) הסיסמאות השמורות על השרת ועל ה domain controller הם לא "salted" – מה שאומר שמחרוזת ראנדומאלית לא נוספת לhashed password כדי להגן עליה עוד יותר. משמעות הדבר היא שאויבים המחזיקים בקוד ה hash לא צריכים את הסיסמא הבסיסית כדי להתאמת. כתוצאה מכך, מערכות היו פגיעות להתקפות brute force.

כאשר הhash של הסיסמא הוא לא salted, זה אומר שאני יכול לעשות brute force ולנסות את מזלי מה שנקרא (תקיפות כאלה יכולות לקחת שעות, אפילו ימים), אבל כאשר הוא salted, גם אם התוקף שם את הסיסמא הנכונה, הוא לא יודע איזה תווים ראנדומאלים נוספו, מה שמקשה עד כמעט בלתי אפשרי לעשות עם brute force

הוספת המחרוזת עוזרת בכך שכאשר מוסיפים תווים או מספרים ראנדומלים לסוף המחרוזת **לפני שעושים לה HASH,** אז גם אם יודעים את הסיסמא המקורית, לא נוכל ליצור את אותו hash, וזה גם יותר מצב שגם אם הסיסמאות הן אותו הדבר, הhash שלהן לא (בגלל התווים הראנדומלים).

עוד מתקפה שהגרסה הראשונה של NTLM (Net-NTLMv1) הייתה פגיעה, היא Pass the Hash (PtH), **בקצרה: מתקפה זו קוראת כאשר התוקף "גונב" את הסיסמא ה hashed ומשתמש בה כדי ליצור סשן משתמש חדש על אותה הרשת.**

**יש לנו את הhash של הסיסמא, זה גם מוסבר בשאלה למטה יותר לעומק.**

איך זה עובד?

התוקף מקבל גישה לרשת המדוברת דרך כל מיני שיטות, לדוגמא Phishing (ממש בקצרה: זה כאשר תוקף מנצל את רגשותיו של אדם אחר, כמו פחד, אמפתיה או חמדנות, כדי לשכנע אותו לשתף מידע אישי או להוריד קובץ זדוני), לאחר שהתוקף מקבל גישה לרשת, הוא משתמש בכלים וטכניקות שונות אשר גורפות (scrape) את הזיכרון הפעיל כדי להגיע לhashes.

לאחר מכן לתוקף יש כמה סיסמאות תקפות מבלי לדעת את הסיסמא המקורית, (מכיוון שNTLM משווה hashes אז אני לא באמת צריך לדעת את הסיסמא המקורית, רק את הhash), כעת התוקף יכול להתחזות למשתמש בכל מיני אפליקציות ולעשות דברים.

הרבה פעמים סוג כזה של התקפה מגיע ל hash harvesting, איסוף מידע על הרבה hashes כדי לקבל הרשאות חזקות יותר או לגשת ליותר מקומות ברשת המדוברת.

Net-NTLMv2 – הגרסה השנייה של NTLM מכיוון שהגיבוב (hash) של הגרסא הראשונה, היה הפיך מידי והיה קל להשיג את הסיסמא המקורית.

Net-NTLMv1 – הגרסה הראשונה של NTLM אשר נחשבת כיום סיכון אבטחתי.

LM HASH – שיטה מדור קודם (legacy) לאחסון סיסמאות במערכות windows.

NTLM HASH – גיבוב קריפטוגרפי של סיסמת המשתמש המשומש בNTLM, זהוי פונקציה חד-כיוונית הממירה סיסמה למחרוזת ייחודית בגודל קבוע.

**NTLM relay**

זוהי אחת מהמתקפות הידועות ביותר על אימות שמבוסס NTLM, בה התוקף משתמש בזהות של הקורבן בלי לדעת את הסיסמא שלו ע"י "שיקוף" של בקשת האימות לשרת אחר.

למה זה מתאפשר?

NTLM לא כולל מנגנון שמחבר בין מקור הבקשה לבין היעד. כלומר אם תוקף ייקח את בקשת האימות, הוא יכול לשלוח אותה הלאה לשרת אחר, והשרת יקבל אותה כאילו היא לגיטימית.

איך זה עובד?

1. התחזות לשרת – התוקף גורם לקורבן לנסות להתחבר לשרת מזוייף (על ידי התקפות שונות, כגון: Email עם קישור SMB, או HTTP מזוייף)
2. קבלת בקשת NTLM מהקורבן – הקורבן שולח את הקשת האימות NTLM Negotiate
3. העברתה לשרת אמיתי (relay) – התוקף משקף את הבקשה לשרת יעד (למשל שרת LDAP,SMB או HTTP בארגון)
4. השרת האמיתי מגיב עם אתגר (challange) – התוקף מקבל את הchallange ושולח אותו בחזרה לקורבן
5. הקורבן עונה עם response – התוקף שולח את הresponse שוב לשרת האמיתי
6. השרת מאמת את זהות הקורבן – השרת מאשר את ההתחברות כאילו הקורבן התחבר בעצמו
7. ביצוע פעולות בשם הקורבן – אם לקורבן יש הרשאות, אפשר לעשות שטויות.
8. **הסבר על Kerberos**

עשיתי במבחן בית.

הערות - אצטרף תמונות

1. **מי זה היוזר krbtgt ומה תפקידו?**

משתמש מיוחד, חשבון משתמש מובנה בAD אשר משומש על ידי KDC (Kerberos Key Distribution), KDC מוסבר במבחן בית, החשבון משומש בשביל להצפין ולחתום על כל כרטיסי הKerberos בתוך דומיין.

המשתמש אחראי על הנפקת כרטיסים מסוג TGT.

אי אפשר להתחבר אליו מכיוון שזה לא חשבון רגיל, זה חשבון מערכת מיוחד שנוצר אוטומטית בעת יצירת הדומיין.

הוא משמש אך ורק לפרוטוקול Kerberos – לחתימה ואימות של TGT

הוא לא אמור להתחבר למחשב, לשרת או להתחבר אינטראקטיבית (אין לו שימוש כזה)

הנה כמה מנגנוני הגנה שמונעים התחברות:

* **החשבון מושבת** – לעיתים קרובות החשבון מסומן כךא זמין להתחברות רגילה (למרות שהוא פעיל)
* **אין הרשאות כניסה** – אין למשתמש הרשאות logon כמו RDP או interactive
* **אין חברות בקבוצה** – הוא לא חבר באף קבוצה שמאפשרת שימוש כלשהו
* **אין צורך בסיסמא** – אין צורת בסיסמא של המשתמש כדי "להשתמש" בו, רק את הhash
* **לא ניתן לשנות את הסיסמא בקלות** – שינוי הסיסמא נעשה ע"י Admin בזהירות רבה (למשל אחרי פריצה)

אבל עדיין אפשר לקחת את הסיסמא שלו עם מתקפה שנקראת Golden ticket (מוסבר באחת השאלות למטה).

אם הצלחתי להשיג את הסיסמא שלו, קודם כל, יש לי את אחד המפתחות הקריפטוגרפים החזקים ביותר בדומיין.

אני יכול ליצור מפתות TGT שתקפים לשנים (Golden ticket), אני יכול לגשת לכל שירות שדורש Kerberos כמו RDP, GPO, SQL, LDAP ועוד.

אני גם יכול להתחמק מזיהוי מכיוון שהכרטיס המזוייף נראה לגיטימי לגמרי, חתום ע"י מפתח חוקי בדומיין.

מה אפשר לעשות אם זה קורה?

אפשר לשחזר את הkrbtgt, שזה אומר לשנות את הסיסמא פעמיים:

1. משנים את הסיסמא, פעם ראשונה כדי לבטל את התוקף של כל הTGT המזוייפים
2. להמתין שפג התוקף של TGT חוקיים
3. לשנות את הסיסמא שוב, כדי לבטל אפשרות להשתמש בhash הישן
4. לוודא שאין שירותים שמסתמכים על Kerberos באופן מיוחד

למה משנים פעמיים?

כאשר מחליפים את הסיסמא של הkrbtgt הוא שומר את הhash של הישן ושל החדש כדי לא לפגוע במשתמשים עם TGT חוקי שנוצרו לפני השינוי.

אז כאשר משנים פעמיים, נשאר hash חדש וישן שלמעשה שניהם "חדשים"

1. **הסבר על התחברות באמצעות כרטיס חכם**
2. כרטיס חכם וקורא כרטיסים

כרטיס חכם, המכיל מיקרו-מעבד ומפתחות קריפטוגרפיים, מוכנס לקורא כרטיסים המחובר למחשב.

1. הזנת סיסמה (PIN)

המשתמש מתבקש להזין מספר זיהוי אישי (PIN) המשויך לכרטיס החכם. קוד הPIN פותח את נעילץ הכרטיס ומאפשר להשתמש בו לאימות.

1. תהליך אימות

מערכת ההפעלה של המחשב, ביחד עם קורא הכרטיסים, מפעילים מנגנון של תגובה לאתגר (challenge-response mechanism)

1. החלפת מפתחות ואימות

הכרטיס החכם משתמש במפתחות קריפטוגרפיים המאוחסים בו בשביל לייצר תשובה לאתגר, התשובה הזו מאומתת לאחר מכן על ידי המחשב מול הנתונים המאוחסים שלו או על ידי שרת. (לדוגמא, אפשר להשתמש במנגנון של AD או בKerberos)

1. אימות מוצלח

לאחר שהאימות הצליח, המשתמש מקבל גישה למערכת ולמשאבים שלה.

שלבים קצת יותר מפורטים:

* 1. **הכנסת הכרטיס החכם** – המשתמש מכניס את הכרטיס החכם לקורא, הכרטיס מכיל תעודה דיגיטלית (X.509) עם מפתח ציבורי ו-Subject שכולל את ה-UPN (שם המשתמש האחיד, לדוגמא: [daniel.cohen@corp.example.com](mailto:daniel.cohen@corp.example.com))
  2. **הזדהות מול ה Domain Controller** – המשתמש מקליד את הPIN של הכרטיס החכם, המערכת שולפת את התעודה מהכרטיס, Windows משתמש בתעודה כדי ליצור PKINIT AS-REQ (קריאה לאימות Kerberos) חתומה עם המפתח הפרטי מהכרטיס
  3. **הDC בודק את התעודה** – הDC בודק את תקופת התעודה (תאריך, שרשרת חתימה, CRL), האם הUPN קיים בAD, האם המשתמש מוגדר ל Smart card logon, אם הכל תקין הDC שולח TGT חתום ללקוח.
  4. **קבלת TGT** – כמו באימות סיסמא, הלקוח מקבל את הTGT ומתחיל לעבוד עם Kerberos

הבדלים עם סיסמא רגילה הנה תמונה חמודה שמצאתי: A screenshot of a computer

AI-generated content may be incorrect.

היתרונות של כרטיס חכם:

* לא ניתן לקחת את הhash כמו בpass the hash
* לא נשמרת סיסמה בזיכרון
* תוקף חייב את הכרטיס הפיזי ואת הPIN
* עמידות גבוהה נגד פישינג

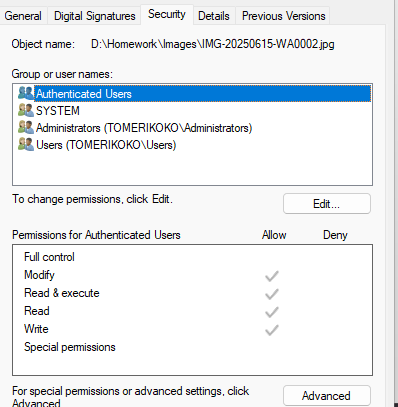
1. **על מה יכול להשפיע סנכרון שעונים לא תקין בין מחשבים?**

סנכרון שעונים לא תקינים יכול להשפיע על פעולות שונות שדורשות תזמון מדויק, כמו העברת נתונים, ביצוע פעולות ברשת, בדיקת אבטחה ורישום איועים, בנוסף, יכולה להיות לו השפעה על תהליכי פיתוח תוכנה, ניתוח נתונים, וכן על דיוק המידע המוצג למשתמש.

מכיוון שKerberos תלוי בזמן (אבל ממש תלוי), כדי שהוא יוכל להגן בפני התקפות "העתקה" (replay). כדי שזה יעבוד הוא בודק את חותמת הזמן של בקשות האימות.

ולכן אם השעון של לקוח מאחר (או מקדים) ביותר מחמש דקות (לדוגמא) אז הבקשה תדחה בגלל חותמת הזמן של הבקשה.

1. **הסבר על הרשאות NTFS והרשאות Share. הסבר על הרשאות על אובייקטים בAD. מה ההבדל ביניהן? איך כל אחת מהן נאכפת?**

הרשאות NTFS (New Technology File System) – NTFS היא מערכת קבצים נפוצה (file system) בעיקר במערכות הפעלה של מיקרוסופט. הרשאות ל NTFS מאפשר שליטה בגישה לקבצים ותיקיות. יש כמה דרכים לנהל את ההרשאות, כמו לגשת לטאב של אבטחה כאשר לוחצים על המאפיינים של תיקייה או קובץ 

יש גם משהו שנקרה ACLs (Access Control Lists), זה מאפשר לאדמינים שליטה מדוייקת על גישה של משתמש לקבצים, איזה הוא יכול ואיזה לא.

הרשאות Share – בגדול עובד כמו הרשאות רגילות של NTFS רק שההבדל הוא שהרשאות share הן רק לקבצים ותיקיות הנמצאות ברשת (לדוגמא קבצים בכונן F, או איך שלא קראת לו אצלך)

דוגמא לפקודה (CMD או powershell):

net share ShareName=C:\SharedFolder /GRANT:Everyone,FULL

הפקודה הזו תיצור שיתוף ברשת בשם “ShareName”, אנחנו משתפים את תיקיית C:\SharedFolder ונותנים לכולם גישה אליה, וגישה לעשות כל מה שרוצים על הקובץ (או תיקייה).

הרשאות אובייקטים בAD – שליטה על איזה משתמשים ו/או קבוצות יכולות לגשת ולשנות אובייקטים בתוך הAD. ההרשאות מגדירות את הרמה של הגישה שיש למשתמשים או קבוצות על אובייקטים אחרים בAD, הם מחליטים על איזה פעולות המשתמש יכול לעשות על אובייקט ספציפי, כמו קריאה, כתיבה, עריכה או מחיקה.

דוגמא להרשאות AD – באחת מהמשימות שהייתי צריך לעשות, הייתי צריך לשלוף מידע מהAD דרך הSAP, פתחו לי את החיבור, רשמתי את הקוד והכל טוב ויפה, אך אבוי! זה לא עובד!

אחרי כמה חקירות הבנתי שהמשתמש ששלף מAD היה בלי הרשאה לגשת למידע הזה, ולכן ביקשתי שיפתחו לו.

איך נאכפות?

NTFS – על ידי מערכת ההפעלה.

בנוגע לShare permissions ו NTFS – מי שיותר מגביל (restrictive) יהיה המנצח כאשר מנסים לגשת למשאב משותף ברשת, לדוגמה: אם בNTFS יש לי "גישה מלאה" ובShare יש לי גישה רק לקריאה, אז יהיה לי גישה רק לקרוא קבצים או תיקיות אשר נמצאים ברשת, על המחשב שלי, תהיה לי "גישה מלאה" כי הShare לא עובד כל המערכת הלוקלית.

בנוגע לאכיפה של הAD – דרך מערכת של access control lists (ACLs) אשר מגדירים מי יכול לגשת ולאן.

ACL – (כמעט בטוח שהסברתי על זה במבחן בית, כי זה קשור לNTLM וkerberos אבל עדיין אסביר בקצרה) – רשימת כללים המגדירים אילו משתמשים או מערכות מקבלים גישה למשאבים ספציפיים ואיזו רמת גישה יש להם

1. **מה זה SID? מה זה GUID?**

SID (Security Identifier) – מזהה ייחודי ובלתי ניתן לשינוי המשמש לזיהוי עקרונות אבטחה כמו משתמשים, קבוצות ומחשבים.

* מבנה – בנויים כסדרה של מספרים בפורמט מסויים כמו: S-1-5-21-xxxxxxxxxx-xxxxxxxxxx-xxxxxxxxxx-yyyy

ה S-1-5 מציין שמדובר בזיהוי אבטחה

זה לא תמיד יהיה 1 ו 5 אך הרבה מאוד מהSID מתחילים ככה מכיוון:

S – מציין שזה SID

1 – גרסת הSID (בדרך כלל 1)

5 – מזהה סידורי של Authority – במקרה הזה, Windows NT

ולכן הרבה מהSID מתחיל בS-1-5 אבל זה לא מחייב.

המספר הבא מציין את רמת הגרסה של הSID

המספרים הבאים מזהים את רשות האבטחה, הדומיין או המחשב, ואת המזהה היחסי RID (Relative identifier) של האובייקט.

נוכל לזהות SID של דומיין אדמין מכייון שיש לו את המבנה הבא:

S-1-5-domain-512

S-1-5 – דיברנו עליו למעלה

Domain – מזהה דומיין, רנדומלי אבל זהה לכל האובייקטים בדומיין.

512 – RID שמייצג את קבוצת Domain Admins

הנה רשימה של כמה RID חשובים:

512 – Domain Admins

513 – Domain Users

514 – Domain Guests

518 – Schema Admins

519 – Enterprise Admins

* יצירה – כאשר משתמש נכנס, ווינדוס יוצר access token (מוסבר שאלה הבאה) אשר מכיל את הSID של המשתמש, הSID של הקבוצה אליה הוא משוייך וההרשאות שלו. הכרטיס הזה משומש לאחר מכן לאימות פעולות של המשתמש.
* שימוש – משמש כדי לנהל גישה למשאבים כמו קבצים, תיקיות ומדפסות. והם קובעים את ההרשאות וההרשאות המשויכות למשתמש או לקבוצה
* SID ידועים מראש – ווינדוס מגדיר כמה SID ידועים המייצגים משתמשים וקבוצות גנריות, כמו "כולם" (Everyone) או "מנהלים" (Administrators)
* AD – בתוך הAD הSID שמור בשדה של ObjectSID

SID History

זה שדה במשתמש או קבוצה בAD שבו נשמרים SIDים ישנים של האובייקט, זה קורה כשמעתיקים או מעבירים משתמש או קבוצה מDomain או Forest אחר, או כשמשנים את מבנה הארגון. המטרה של שדה זה היא לאפשר למשתמש לגשת למשאבים שמאופשרים לSID הישן, גם אחרי המעבר.

תוקף יכול לנצל את זה לרעה בכך שהוא מוסיף לעצמו SID של קבוצה עם הרשאות חזקות לhistory ובכך יהיה לו את ההרשאת של הקבוצה גם מבלי להיות חבר בה.

GUID (Globally Unique Identifier) – גם מוכר כ UUID (Universally Unique Identifier), זה ערך בגודל של 128 ביט (128-bit value) המשמש לזיהוי יישויות ייחודיות במערכת, במיוחד במחשוב וברשתות.

* מבנה – מספרים שלמים של 128 סיביות (128 bit), מה שמספק מספר גדול מאוד של ערכים אפשריים, מה שמפחית את הסיכוי להתנגשויות. מה שהוא טכנית 128-bit binary value מיוצג הרבה פעמים כמחרוזת של 32 תווים הקסדצימלים, כגון: 30dd879c-ee2f-11db-8314-0800200c9a66
* יצירה – יש כמה וכמה דרכים בהם מיצרים GUID, אסביר עליהם בקצרה:

**Time-Based –** משלב את החתימת זמן, רצף השעון ומזהה צומת (בדרך כלל MAC address)

**Random** **–** רוב המספרים הבונים את הGUID הינם ראנדומאלים, 6 סיביות (6 bits) שמורים לגרסה ווריאציה. שאר ה122 סיביות ממולאות במספרים ראנדומאליים.

**Name-Based –** גיבוב של מזהה מרחב שמות (namespace) ושם, גרסא 3 משתמשת בגיבוב (hashing) MD5 וגרסה 5 משתמשת בSHA-1.

**Content-Based –** מיוצר על ידי גיבוב התוכן של קובץ. קבצים עם אותו התוכן יקבלו את אותו הGUID, שימושי כאשר רוצים למצוא קבצים זהים.

* שימוש – משמש באופן תדיר כמפתח ראשי בטבלאות במסד נתונים, בAD יש לכל אובייקט GUID משלו, וכמובן עוד כמה שימושים, לא ארשום כאן את כולם.

1. **מה זה Access Token?**

אישור המאפשר לאפליקציה גישה למשאבים מטעם משתמש או מטעם עצמה, מהלי שיהיה צורך לאמת מחדש שוב ושוב. משומש כמו מפתח דיגיטלי, מאפשר גישה למשאבים מסויימים לזמן מוגבל (לרוב שעה, אבל ניתן להגדיר את זה). משומש לרוב עם web API ומונפק על ידי שרת הרשאה (authorization server) לאחר אימות מוצלח.

איך זה עובד

* + משתמש או יישום מאמתים מול שרת הרשאות, בדרך כלל האמצעות שם משתמש/סיסמה.
  + לאחר אימות מוצלח, שרת ההרשאה מנפיק אסימון גישה.
  + לאחר מכן, היישום משתמש באסימון גישה (access token) כדי לגשת למשאבים מוכנים מטעם המשתמש.
  + שרת המשאבים מאמת את הaccess token כדי לאמת את האותנטיות שלו ואת היקף ההרשאה.

בתוך access token יש הרבה מידע, ביניהם:

* SID של משתמש
* SID של קבוצות
* Privileges
* Owner SID
* Restricted SID – במקרים מסויימים כדי לצמצם גישה
* Session info
* Integrity level – משתמש רגיל, מנהל מערכת או שירות

סוגים – ישנם שני סוגים כיום:

JWT (JSON Web Token) – פורמט נפוץ אשר מכיל מידע לגבי המשתמש והרשאותיו (claims)

Opaque Tokens – מחרוזות שאינן גלויות ללקוח ומטופלות על ידי שרת ההרשאות.

למשתמש יכול להיות יותר מaccess token 1, הנה כמה מקרים:

* **משתמש מריץ תהליכים שונים ברמות הרשאה שונות** – למשל תהליך אחד רץ רגיל ואחד כאדמין, כל אחד מהם יקבל access token שונה עם סט הרשאות שונות.
* **Impersonation** – שירות או תהליך (למשל IIS) מריץ קוד בשם המשתמש – נוצר access token חדש זמני עבורו (Impersonation Token)
* **Session נפרד (RDP, runas, remote login)** – משתמש שמחובר מקומית וגם דרך RDP מקבל access token שונה לכל session

הaccess token נשמר בRAM.

\*\*\*עבדתי וגם יצרתי access tokens, אם תרצי, אשמח להסביר לך על מה עשיתי

[להסביר]

1. **מה זה integrity level?**

מנגנון אבטחה המגדיר רמת אמון הקשורה לתהליך או לאובייקט, כמו קובץ או תוכנה.

הרמות מיוצגות לעתים קרובות כערכים מספריים או ערכים המחרוזת (Low, Medium, High, System)

למה זה חשוב?

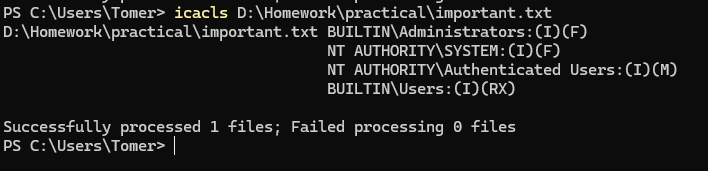
מניעת שינוי בלתי מורשה של נתונים, ומבטיחות שהמידע יישר מדוייק ועקבי.

עוד תוסף לאבטחה כדי להקשות על תוקפים, גם אם יש לי גישה למערכת, יכול להיות שזה רק בLow ולכן אני לא יכול לגשת למה שחשוב באמת.

דוגמא: אם הדפדפן רץ על רמה medium הוא לא יוכל לגשת לקבצים שמעל medium (כמו High או system).

מונע שינוי בכך שתהליך ברמת integrity נמוכה לא יכול לכתוב או לשנות אובייקט ברמת integrity גבוהה, גם אם יש לו הרשאות בNTFS.

ניתן להשתמש בפקודה icacls כדי לראות את הintegrity.



A screenshot of a computer

AI-generated content may be incorrect.

1. **ברצוננו לתת הרשאות ליצירת יוזרים ואיפוס סיסמאות בלבד לחוליית מחשב אשדוד על כל המשתמשים שלהם. איך נעשה את זה? (לתת הרשאה מדוייקת ואותה בלבד)**

צריך לבדוק עם נועה אם אני אמור לדעת איך הם עובדים...

נראה שרצית לשאול אותי ☺ לא קריטי באמת חוליית אשדוד אז אני אנסח את השאלה קצת אחרת – ברצוני לתת הרשאות ליצור יוזרים ואיפוס סיסמא בלבד לחוליה רק על המשתמשים שלהם. כלומר לא לתת לחוליית אשדוד הרשאות על משתמשי בצת.. וגם רק את ההרשאות הרלוונטיות ולא FULL CONTROL. תמצא את ההרשאה המדובר ותסביר איך היית עושה את זה.

[לעשות]

1. **כיצד ניתן לראות את ההרשאות של משתמש מסויים על OU ספציפי?**

יש כמה דרכים, כמו כלים שמיועדים לכך כמו Netwrix Auditor או ADACLScan,

יש גם דרך הUI במשהו שנקרה Active Directory Users and Computers (ADUC)

לא אסביר על זה כי אפשר בקלות למצוא איפה לוחצים באינטרנט....

אבל כן אדבר על דרך שקצת יותר מדברת אלי, דרך powershell.

$ouDN = "OU=Sales,DC=example,DC=com"  
$userName = "JohnDoe"  
$userDN = (Get-ADUser -Identity $userName).distinguishedname  
$acl = Get-Acl "AD:$ouDN"  
$userPermissions = $acl.Access | Where-Object {$\_.IdentityReference -like $userDN}  
  
foreach ($permission in $userPermissions) {  
 Write-Host "User: $($permission.IdentityReference)"  
 Write-Host "Type: $($permission.AccessControlType)"  
 Write-Host "Rights: $($permission.ActiveDirectoryRights)"  
 Write-Host "---"  
}

הסקריפט הזה מחזיר את כל ההרשאות שיש ל JohnDoe בתוך ה Sales OU ומדפיס אותם לconsole.

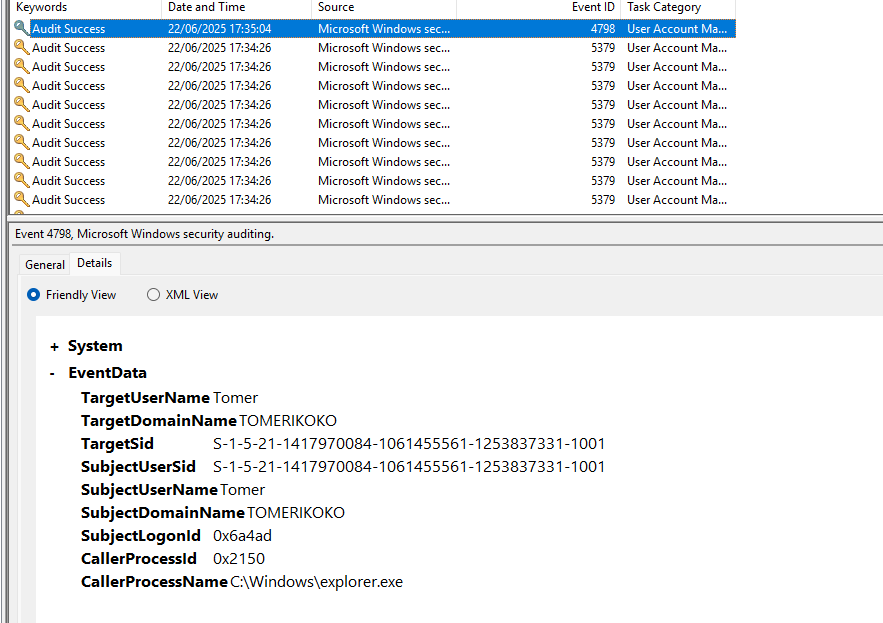
1. **למה להשתמש בלשונית Advanced בתוך Security TAB? ואילו דברים מעניינים נוכל ליישם שם?**

הלשונית Advanced מספקת גישה להגדרות הרשאה מפורטות ומיוחדות יותר מאשר ב Security. לדוגמא לנהל הרשאות שאינן גלויות או ניתנות להגדרה ישירות מהלשונית הרגילה.

אפשר לעקוב (סטוקר סטייל 😎) אחר מי ניגש לאיזה קובץ ומתי.

כדי לעקוב אחרי מישהו יש לעשות את הדברים הבאים:

1. פותחים את הsecurity בקובץ או תיקייה שרוצים לעקוב אחריה
2. פותחים Advanced
3. הולכים לAuditing
4. לוחצים על Add
5. בוחרים principal
6. בוחרים על איזה פעולות רוצים שיופיע המידע
7. וזהו! עכשיו כל פעם שמישהו נכנס לקובץ אפשר לראות את זה בEvent Viewer (אצלי זה success אבל אפשר לעשות את זה כfail)



עדיין קצת קשה לראות איזה קובץ זה (אם אפשר בכלל) אבל זה עדיין משהו, מניח שאם אשחק עם זה עוד קצת יהיה אפשר, תמיד אפשר לעשות windows services שמקשיב לקובץ ורושם לבדו לevent log עם סקריפט של powershell

אפשר להעביר בעלות של קובץ או תיקייה למשתמש אחר או קבוצה.

אפשר לחסום למשתמש או לקבוצה מלמחוק קובץ אבל עדיין לאפשר להם לראות אותו.

אפשר לראות בדיוק מה משתמש יכול או לא יכול לעשות, גם אם יש הרשאות מתנגשות.

הוספת הרשאות על תתי-קבצים/תיקיות בלבד, למשל הרשאה שקיימת רק על קבצים בתיקייה ולא על התיקייה עצמה

הוספת הרשאות מדוייקות, כמו רק Append data או delete

ניהול ירושה, לבטל או להפעיל ירושה של הרשאות מתיקיות אב

בגדול משתמשים בזה כאשר רוצים שליטה יותר טובה ומדוייקת על הרשאות שיש למשתמשים או קבוצות על קבצים, או אפילו למטרות של מעקב.

1. **קיים המצב הבא – אסף נמצא בתוך הקבוצה “Cyber” שיש לה full control על התיקייה של אורי. אורי שינה את ההרשאות על התיקיה שלו ושם שלאסף יהיה deny על כתיבה. מה בתכלס אסף יוכל לעשות?**

מכיוון שהסברנו למעלה איך הרשאות עובדות ונאכפות (מה שיותר חוסם, הוא הלוקח) זה אומר שמה שאסף יוכל לעשות זה:

לקרוא קבצים, לרשום הרשאות (אם זה לא נחסם דרך הDENY), למחוק קבצים, לבצע כל פעולה אחרת שאין עליה deny

בגדול הוא רק לא יוכל לרשום לתיקייה. ושאר הפעולות תלויות בdeny שאסף קיבל, כל פעולה בלי deny הוא יוכל לעשות.

כדי לקחת בעלות על הקובץ הוא צקיך הרשאה בשם Take ownership, כל עוד אורי לא עשה DENY להרשאה או שאסף הוא אדמין, הוא יוכל לקחת בעלות.

1. **מה ההבדל בין הפעולות שאפשר לעשות על משתמש שמוצאים דרך חיפוש. לבין משתמש שמוצאים אותו במקום ההיררכי שלו ב AD?**

כאשר מחפשים משתמש דרך החיפוש מקבלים **תצוגה ממוקדת של האובייקט** בלי קשר למיקומו ההירכי (OU), הפעולות המותרות מצומצמות יותר, לדוגמא לא נוכל להעביר אותו לOU אחר, לא נוכל ליצור אובייקטים חדשים לידו ולפעמים אפילו שינוי של הרשאות מסויימות יהיה חסום. ולמה זה? בגלל שהADUC לא יודע באיזה OU מדובר, ולכן רואים רק את האובייקט ולא את המיקום שלו בעץ.

כמובן שאם מוצאים אותו במקום ההיררכי שלו, תהיה לנו גישה מלאה לעשות הכל.

[לעשות]

1. **מהם Containers. ובמה הם מיוחדים?**

הם אובייקט בתוך הAD שמכיל אובייקטים אחרים כמו משתמשים, קבוצות, מחשבים וכו'.

* + הם אובייקטים "קשיחים", מוגדר מראש ולא ניתן לשנות אותו.
  + לא ניתן להחיל עליו Group policy (GPO)
  + לא ניתן לבצע עליו האצלת סמכויות
  + אי אפשר ליצור containers חדשים – רק OU

מה שמיוחד בו הוא מובנה מראש ע"י מערכת ההפעלה, משמש כברירת מחדל להצבת אובייקטים חדשים והוא קבוע במבנה, לא ניתן לשכפול או יצירה.

לסיכום: זהו אובייקט סטטי, מובנה, שלא ניתן להתאים אישית, ואינו תומך במדיניות AD רגילה כמו OU.  
נוח כברירת מחדל, אבל לא גמיש – ולכן לא מתאים לניהול ארגוני אמיתי.

1. **הסבר מהו adminSDHolder**

אובייקט מיוחד בAD שנמצא בתוך הDomain System Container, משמש כתבנית להרשאות בקבוצות מוגנות וחבריהן.

התפקיד העיקרי הוא לאכוף תצורת אבטחה עקבית עבור קבוצות קריטיות וחבריהן (members)

יש תהליך שנקרא SDProp (Security Descriptor Propagation) , שהו תהליך שרץ אוטומטית וכברירת מחדל, כל 60 דקות (שעה) על הdomain controller שמחזיק בתפקיד האמולטור של PDC. (Primary Domain Controller)

איך זה עובד?

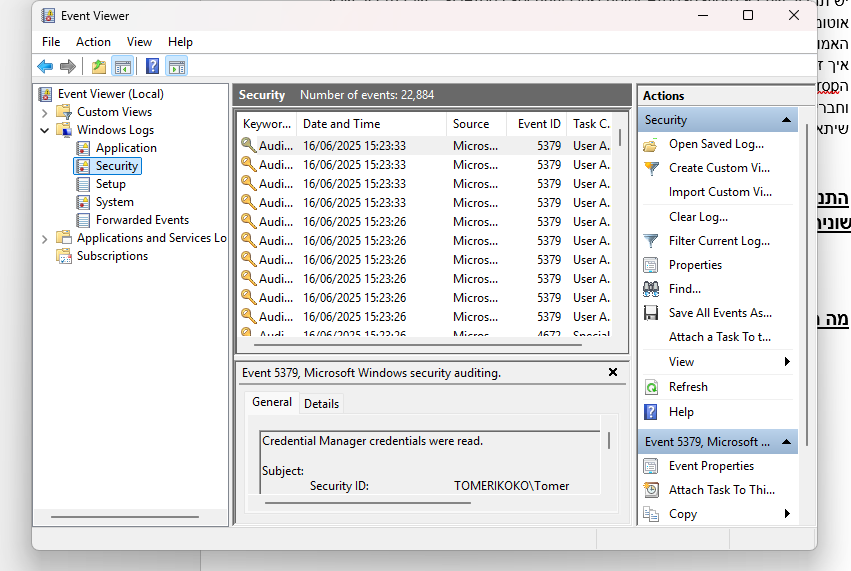
הSDProp משווה את הACL של ה adminSDHolder עם הACL של הקבוצה המוגנת וחבריה, אם זוהה שינוי, הSDProp מאפש את ההרשאות של האובייקטים המוגנים כדי שיתאימו להרשאות שהוגדרו הadminSDHolder.

זה מעניין תוקף מכיוון שאם הוא מצליח לשנות את ההרשאות על האובייקט הזה הוא משגי שליטה מתמשכת על חשבונות קריטיים.

דוגמא:

* 1. התוקף משיג הרשאות גבוהות באופן זמני.
  2. הוא עורך את הACL ומוסיף לשם הרשאות למשתמש שהוא שולט עליו.
  3. בכל שעה – AD באופן אוטומטי מעתיק את ההרשאות האלה לכל חשבון רגיש.
  4. גם אם IT ינסו לשנות את ההרשאות ידנית, הAD ידרום אותם אחרי שעה!

1. **התנתקתי מהמחשב ע"י ביצוע logoff. אראה אירוע על כך ב eventlog בלשונית של?**

****

Windows logs -> Security

יותר ספציפי: יש לחפש את הEvent ID 4634, An account was logged off

אומנם אני לא בבסיס כדי לבדוק את זה, אבל מצאתי שאם המשתמש לחץ על logoff או יציאה, הלוג 4647 יירשם. User initiated logoff, הלוג הזה מתעד את הפעולה הידנית של המשתמש. בניגוד לזה בלוג 4634 יכול להרשם בלי 4647 לפניו אם המחשב התכבה, עשה restart או בגדול כל דבר שגורם לsession של משתמש להסתיים בלי שהמשתמש עשה זאת בפועל.

1. **מה ההבדל בין Features ל – Roles? ותן דוגמא למספר Features.**

Roles – הם רכיבים עיקריים של מערכת ההפעלה, תפקידים שהשרת ממלא בארגון, שרת יכול לבצע כמה roles בו זמנית לדוגמא כמו שרת AD שהוא גם שרת DNS (צריך כדי שהAD יעבוד כמו שצריך). אבל לא תמיד רצוי לעשות דבר כזה מכיוון שכל role מושך משאבים או פותח סיכוני אבטחה, וגם השרת לא בטוח יעמוד בעומסים גבוהים או בארגונים גדולים.

Features – כלים משלימים או יכולות תומכות שאינן תפקיד מרכזי, אך מוסיפות פונקציונליות לשרת.

דוגמאות לroles:

* + Web Server (IIS) – מציג אתרי אינטרנט (התעסקתי גם עם כמה בשירות)
  + DNS Server – כמו שמוסבר בשאלה אחת, פותר שמות ברשת.
  + DHCP Server – מקצה כתובות IP אוטומטית ללקוחות

ועוד כמובן

דוגמאות לfeatures:

* + NET Framework 3.5 / 4.x – מאפשר להריץ אפליקציות מבוססות .NET
  + Telnet Client – כלי בדיקה לרשת
  + BitLocker Drive Encryption – הצפנת דיסקים  
    ועוד כמובן.

1. **ברצוני לפרמט מחשב ולהכניס לדומיין (אצלנו בחיל). פרט את השלבים החיצוניים.**

[מניח שאם רשום אצלנו בחיל אז זה קצת שונה, בינתיים אשאיר ריק כי אין לי כרגע את מי לשאול]

1. **במה שונה ארכיטקטורת הדומיין ברשת סודית ביותר? פרט.**

[חסר לי]

1. **מנה לפחות 2 כלים איתם מנהלים את הDomain ותן הסבר קצר על כל אחד מהם.**

**ADUC (Active Directory Users and Computers)** – כלי גרפי המאפשר ניהול של אובייקטים בתוך ה Active directory, כמו משתמשים, קבוצות, מחשבים ויחידות.

מה אפשר לעשות איתו?

* + יצירה, מחיקה ועריכה של משתמשים וקבוצות
  + העברת אובייקטים בין הOU
  + הגדרת הרשאות ואפיון תכונות משתמשים
  + נעילת/שחרור חשבונות.

**GPMC (Group Policy Management Console)** – ניהול מדיניות קבוצתית (Group policy) – אוסף של הגדרות שמוחלות על משתמשים ומחשבים בדומיין.

מה אפשר לעשות איתו?

* + יצירה ועריכה של GPO
  + קישור מדיניות ל-OU, דומיינים ואתרים
  + הגדרת הגבלות והרשאות, לדגומא: חסימת גישה ל-USB, הגדרת סיסמאות חובה, ניהול עדכונים ועוד.

1. **מהו DNS ולמה הוא משמש? הסבר על כל תהליך הדרדור התרגום השמות (בסביבת Microsoft, לא DNS עולמי)**

יש גם טיפה הסבר בשאלה אחת.

**DNS** – Domain Name System, תפקידו לתרגם שמות מתחם (Domain Names) לכתובות IP, כלומר להפוך נגיד את זה: server01.domain.local ל: 192.168.1.10

תהליך הדרדור פועל בשלבים מסויימים, **אם שלב אחד נכשל או ממשיך לבא אחריו:**

1. **Host Cache** – בודק בזיכרון המקומי אם התרגום כבר קיים מהעבר (Cache)  
   אם כן – מחזיר את התוצאה מיידית.
2. **קובץ hosts** – בודק בקובץ " C:\Windows\System32\drivers\etc\hosts"  
   אם השם נמצא בקובץ – משתמש בן (בקובץ זה רק תרגומים סטטים, שיחקתי איתו בעבר)
3. **שאילתת DNS** – אם השם לא נמצא מקומית, שולח שאילתת DNS לשרת ה-DNS המוגדר במערכת. ה-DNS עובר לרוב ברשת באירגונית ולא פונה לרשת.
4. **NetBIOS/WINS** – אם השם לא נמצא בDNS, ייתכן שהמערכת תנסה להשתמש בפרוטוקול NetBIOS או שרת WINS כדי לתרגם שמות קצרים (legacy)
5. **Broadcast** – כניסיון אחרון, המחשב עלול לשדר לרשת שאלה כמו: "מי זה server01?" (רק בשמות קצרים וללא DNS)
6. **איך מחשב נרשם לdns? תאר את התהליך מהרגע שהמחשב נכנס לרשת.**

כאשר מחשב מצטרף לרשת מבוססת Windows (למשל דומיין של Active Directory), הוא עובר תהליך רישום מול DNS על מנת שמחשבים אחרים יוכלו לאתר אותו.

* 1. **קבלת כתובת (DHCP) IP**

קודם כל מבצעים בקשת DHCP כדי לקבל כתובת IP, שער ברירת מחדל (gateway), כתובת DNS וכו'

* + המחשב שולח בקשת DHCP
  + שרת ה-DHCP מחזיר כתובת IP ומידע נוסף, כולל כתובת של שרת DNS.
  + המחשב מקבל ומגדיר את עצמו עם המידע הזה
  1. **רישום רשומות (Dynamic DNS Registration) DNS**

כעת, המחשב רוצה להירשם ב-DNS כדי שמחשבים אחרים יוכלו למצוא אותו על פי שם.

* + המחשב שולח בקשת Dynamic DNS Update לשרת ה-DNS, כדי לרשום את עצמו.

הבקשה כוללת: שם מחשב מלא וכתובת הIP שהוא קיבל

* + שרת ה-DNS בודק אם יש הרשאה לעדכון (לרוב רק מחשבים שמאומתים בדומיין יכולים)
  + אם כן, נרשמת רשומת A (Address Record) שמתאימה בין השם ל-IP
  + במקביל, המחשב או שרת DHCP (תלוי בקונפיגורציה) יכול לרשום גם רשומת PTR – שמאפשרת תרגום מכתובת IP לשם.
  1. **רענון תקופתי**
  + המחשב שולח מידי פעם (24 שעות כברירת מחדל) בקשת רישום מחדש כדי לרענן את הרשומה ולמנוע ממנה להמחק.
  + אם המחשב שינה IP – הוא יעדכן את הרשומה בהתאם.

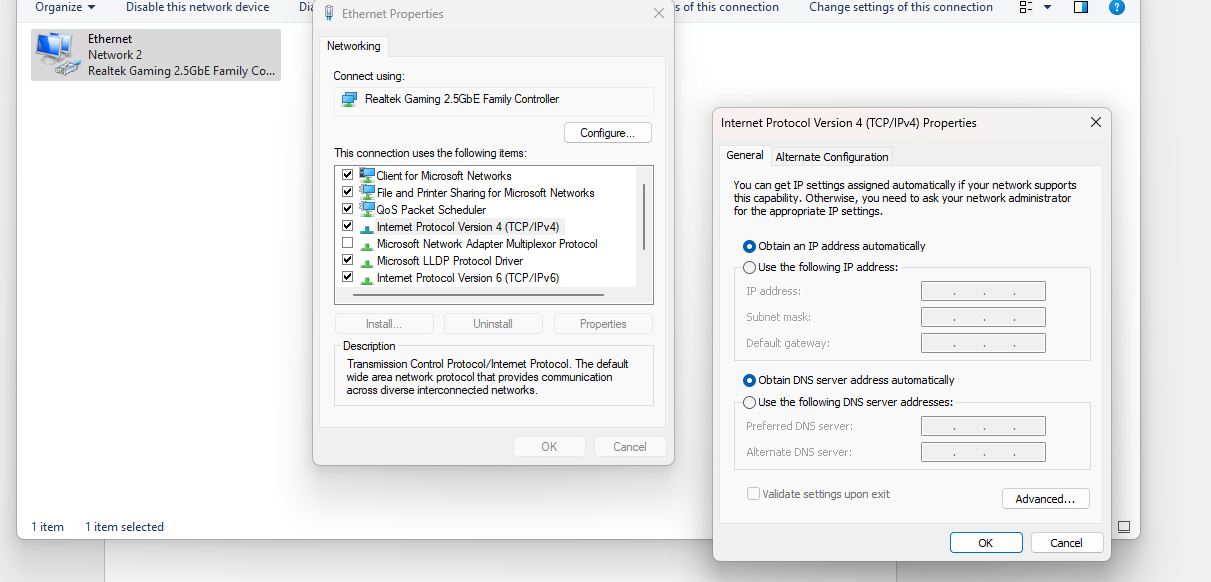
1. **איזה פקודה רושמת מחשב לdns?**

ipconfig /registerdns

פקודה זו גורמת למחשב לשלוח בקשת Dynamic DNS Update לשרת DNS

שימושית במיוחד אם שינית כתובת IP, שם מחשב, או שאת חודשת שהרשומה לא קיימת.

1. **ברצוננו לקבל כתובות משרת dns שמחוץ לארגון שלנו. איך אפשר לעשות את זה. ציין שתי דרכים ואת דרך עבודתם.**
   1. **הגדרה ידנית של שרת DNS חיצוני בclient**
   * בכל מחשב (או נתב) מגדירים ידנית כתובות של שרתי DNS חיצוניים



ניתן להגדיר את זה כאן, בהגדרות של החיבור רשת של המחשב.

* + כשהמחשב מנסה לפתור שם (כמו [www.google.com](http://www.google.com)), הוא פונה ישירות לDNS החיצוני, בלי לעבור דרך הDNS הארגוני.

**יתרונות:**

* + פשוט ויעיל
  + לא תלוי בתצורת ה-DNS הפנימי של הארגון.

**חסרונות:**

* + לא רואה שמות פנימיים של הארגון
  + דורש הרשאה ליציאה על פורט 53 (DNS) החוצה – לא תמיד מותר בפיירוול הארגוני.
  1. **שרת DNS פנימי מבצע Forwarding ל-DNS חיצוני**

שרת ה-DNS של הארגון מוגדר לבצע forward לבקשות שהוא לא מזהה – לשרתי DNS חיצוניים. וזה פועל כך:

* משתמש שואל שאלה לDNS (למשל [www.google.com](http://www.google.com))
* שרת הDNS הארגוני לא מוצא תשובה מקומית.
* הוא מפנה את הבקשה לשרת DNS חיצוני (למשל 8.8.8.8, גוגל)
* התשובה מוחזרת למשתמש דרך השרת הארגוני.

**יתרונות:**

* מאפשר שליטה מרכזית.
* שומר על פרטיות ארגונית (המשתמשים לא פונים ישירות החוצה)
* מאפשר גישה גם לשמות פנימיים וגם חיצוניים.

**חסרונות:**

* דורש הגדרה בשרת DNS ארגוני
* תלוי ביציבות ה-forwarder

1. **מה ההבדל בין reverse lookup zone ל forward lookup zone.**

בקצרה: forward lookup zone זה כאשר ממירים שם דומיין לכתובת IP וreverse lookup zone זה כאשר ממירים ip לשם דומיין.

עכשיו אסביר בקצרה על איך כל אחד עובד:

1. **Forward lookup zone**

כאשר מקלידים כתובת כמו [www.google.com](http://www.google.com) ה-DNS מחפש רשומת A (Address Record) באזור החיפוש הקדמי (Forward lookup zone) ומחזיר את כתובת ה-IP המתאימה.

סוגי רשומות נפוצים:

* + A – שם ל-IPv4
  + AAAA – שם לIPv6
  + CNAME – שם לשם אחר
  + MX – שרת דואר
  + NS – שרתי DNS של הזון

1. **Reverse lookup zone**

אם יש לי לדוגמא 192.168.1.10 ואני רוצה לדעת מי זה – תתבצע שאילתת PTR record באיזור החיפוש ההפוך (Reverse lookup zone) שמתאים לIP הזה, והשרת יחזיר את שם המחשב למשל: server1.corp.local

* + מבוצע באמצעות שמות דומיין מיוחדים כמו: in-addr.arpa.10.1.168.192 (לכתובת IPv4)

סוג רשומה:

PTR – כתובת IP לשם (Pointer Record)

1. **איזה סוגי zone של dns קיימים? מה ההבדל ביניהם?**

תכף אפרט על כל הסוגים, אבל בגדול, ההבדל ביניהם בה לידי ביטוי ב:

* + צורת אחסון
  + האם ניתן לעריכה
  + שימוש עיקרי

עכשיו אפרט:

* 1. **Primary Zone**
  + זה הzone הראשי שבו נמצאות רשומות DNS המקוריות.
  + ניתן ליצור, לערוך ולמחוק רשומות DNS בו.
  + נשמר בקובץ או במסד נתונים מקומי (או בActive Directory אם משולב)
  1. **Secondary Zone**
  + עותק לקריאה בלבד של Primary Zone אחר.
  + מכיל עותק זהה של ה-Primary Zone
  + מתעדכן אוטומטית דרך zone transfer (מוסבר שאלה הבאה) מהשרת הראשי.
  + מיועד לגיבוי, איזון עומסים, או רזולוציה קרובה גיאוגרפית
  + לא ניתן לערוך אותו ישירות (הוא לקריאה בלבד כמו שרשום למעלה)
  + שימושי כאשר רוצים DNS נוסף בלי לשכפל את הנתונים ידנית
  1. **Stub Zone**
  + שומר רק הפניות – לא את כל הרשומות
  + מכיל רק רשומות SOA, NS, ו-A שמצביעות על שרתים שמארחים את ה-Zone המקורי
  + לא כולל את כל הרשומות
  + מיועד כדי לאפשר הפניה יעילה בין אזורים שונים.
  + שימושי במיוחד בסביבות עם דומיינים שונים שיש ביניהם אמון (Trust), למשל בארגון עם כמה אתרי Active Directory
  1. **Active Directory-Integrated Zone**
  + Zone שמאוחסן ומשוכפל דרך Active directory
  + נשמר במסד הנתונים של הAD
  + נתמך רק על שרתי DNS שמריצים Domain controller
  + נתמך ע"י replication אוטומטי בין DC-ים
  + מאפשר עדכון דינמי מאובטח
  + מקל על ניהול DNS בארגון מבוזר, עם יתרונות אבטחה וזמינות.

1. **מהו zone transfer? מה זה dynamic updates בzones מה ההבדל ומה מומלץ?**
   1. **Zone transfer**

תהליך שבו שרת DNS מעביר עותק שלם של Zone (כל הרשומות) לשרת DNS אחר.

שימושים:

* + עיקר השימוש הוא ב Secondary zone – מוסבר למלא
  + משמש לרפליקציה בין שרתי DNS (גיבוי, איזון עומסים וכו)

סוגים:

* + AXFR – מעביר את כל הZone מההתחלה.
  + IXFR – מעביר רק שינויים שנעשו מאז הפעם האחרונה.

אבטחה:

חשוב להגביל Zone transfer רק לשרתים מורשים, אחרת תוקף יכול לקחת את כל שמות הדומיין.

* 1. **Dynamic Update**

מאפשר למחשבים (לרוב חברי דומיין) לעדכן רשומות DNS האופן אוטומטי – לדוגמא, כאשר הם מקבלים כתובת IP חדשב או מצטרפים לרשת.

שימושים:

* + בשילוב עם DHCP – מחשבים יכולים לרשום את עצמם ל-DNS
  + בשרתים מבוססים Active Directory – עוזר לרישום רשומות של Domain Controllers, שרתים, מחשבים ועוד.

סוגים:

* + Non-secure Dynamic updates – כל מחשב יכול לשלוח בקשה לרישום (לא מאובטח)
  + Secure dynamic updates – רק מחשבים מאומתים בדומיין יכולים לרשום/לשנות רשומות (מאובטח ומומךץ)

**מה מומלץ?**

אם אנחנו בסביבת AD אז מומלץ להשתמש בActive Directory Integrated Zone עם Secure dynamic Zone

עדיף להשתמש ב Zone transfer רק לשרתים שמוגדרים כSecondary

1. **איך ניתן לתשאל שרת DNS?**

ניתן לתשאל DNS בעזרת כלים נפוצים כמו: dig (Linux) , nslookup (CMD או POWERSHELL), או host (POWERSHELL)

קצת דוגמאות:

nslookup www.google.com 8.8.8.8

dig @8.8.8.8 example.com A

dig -x 8.8.8.8 (זה תשאול הפוך)

Resolve-DnsName example.com -Server 8.8.8.8 -Type A (powershell)

\*\*\*אם צריך אפרט כאן יותר אבל לא יודע מה עוד אפשר לפרט כאן באמת, זו שאלה די סגורה 😅

1. **יש לנו שרת בשם MAOR ושרת בשם MATAN. אנחנו רוצים שכאשר יפנו ב DNS לשרת MAOR אז הDNS יפנה אותנו לכתובת IP של MATAN. איך נעשה את זה כמנהלים? איך נעשה את זה כתוקפים?**

\*\*\*הסברתי למעלה איך כל הרשומות האלה עובדות, אז לא אסביר כאן, נראה לי בזבוז זמן

\*\*\*לגבי הסבר של תוקפים, אני יכול להכנס לכל אחת מהאופציות שנתתי לעומק, אבל אני לא בטוח שזה הכרחי

**מנהלים:**

יש כמה דרכים לעשות את זה:

1. **שימוש ברשומת CNAME**

בשרת הDNS ניצור רשומת CNAME המצביע על שם אחר, לדוגמא:

MAOR CNAME MATAN.domain.local.

1. **שימוש ברשומת A עם אותו כתובת IP**

אפשר פשוט ליצור רשומת A כפולה, לדוגמא:

MAOR A 192.168.1.20 ← כתובת של MATAN

MATAN A 192.168.1.20

איפה מגדירים את זה?:

ב Windows Server DNS Manager, בוחרים Zone, מקש ימני ואז new alias (cname) או new host (A).

**כתוקפים:**

יש כמה דרכים שמצאתי, אסביר על כולם:

1. **הרעלת DNS (DNS Spoofing / Poisoning)**

בגדול זה אומר שהתוקף מזייף תשובות DNS לקורבן, לדוגמא הקורבן שואל על MAOR, התוקף שולח תשובה מזויפת על כתובת IP של MATAN.

* + מנצל חולשות בשרת DNS או מחליף תגובות לפני שהתגובה האמיתית מגיעה
  + נפוץ ברשתות לא מאובטחות (לדוגמא wifi פתוח)

תנאים:

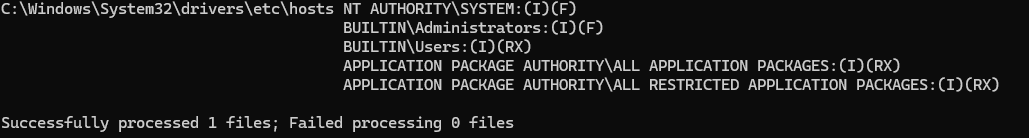
* 1. **גישה לתעבורת DNS או לשרת DNS** – תוקף צריך להיות מסוגל ליירט או לשלוח תגובות DNS, או להשפיע על שרת DNS רגיש/פגיע/לא מוגן
  2. **היעדר אימות תגובות DNS** – מערכות DNS סטנדרטיות לא מאמתות שהתגובה באה מהשרת האמיתי, תגובה של DNS מבוססת על: מזהה, כתובת IP של שרת ותזמון נכון. אם תוקף מנחש נכון את המזהה הוא יכול להזריק תגובה לפני השרת האמיתי
  3. **שרת DNS או לקוח עם cache לא מוגן** – הרעלת DNS נועד לחדור את הCache של DNS Resolver, ברגע שהרשומה המזויפת נשמרת בcache כולם שמבצעים query יקבלו אותה.
  4. **ללא DNSSEC** – אם הדומיין מאובטח באמצעות DNSSEC, כל תגובה צריכה להיחתם דיגיטלית, אם DNSSEC לא מופעל, לא ניתן לוודא שהתגובה לגיטימית.

1. **שינוי קובץ hosts המחשב הקורבן**

כמו שהסברתי למעלה, יש קובץ על המחשב בנתיב C:\Windows\System32\drivers\etc\hosts

ששומר על כל הכתובות והמיפויים הסטטיים של המחשב הזה ספציפית, תוקף יכול באמצעות תוכנה זדונית (או ידנית אם יש לו גישה) לשנות את הקובץ שיתאים למה שאנחנו רוצים.

כמו שאפשר לראות בcommand המגניב והחדש שלמדתי:



למשתמשים רגילים אין אופציה לשנות את הקובץ (יש להם רק read & execute) וצריך להיות מנהלים כדי לשנות את הקובץ (יש להם גישה מלאה). או SYSTEM.

1. **MITM (Man-In-The-Middle)**

התוקף משתלט על התקשורת בין הקורבן לDNS ומחזיר תשובה מזויפת.

1. **מה ההבדל בין משתמש לבין contact? ומה משתמש יכול לעשות שcontact לא יכול?**

בקצרה:

משתמש (כמו: [tomer.mor@company.local](mailto:tomer.mor@company.local)), יכול להתחבר למחשב, להפעיל מערכות, לקבל הרשאות לקבצים, להפעיל יישומים.

Contact (כגון: [tomer.mor@gmail.com](mailto:tomer.mor@gmail.com)) לא יכול להתחבר, אבל יכול להופיע בספר הכתובות ולקבל העתקי מיילים (forwarding)

הנה טבלה שמצאתי שמסבירה מה משתמש יכול לעשות שcontact לא.

| **פעולה** | **Contact** | **User** |
| --- | --- | --- |
| לבצע Login לתחנה | ❌ | ✅ |
| לשנות סיסמה | ❌ | ✅ |
| לגשת לשיתופים/שרתים | ❌ | ✅ |
| לקבל Group Policy | ❌ | ✅ |
| לבצע הפעלה של מערכת | ❌ | ✅ |
| לפתוח session (RDP, VPN וכו') | ❌ | ✅ |

1. **הסבר מהו Kerberos double hop.**

אתחיל בהסבר על Kerberos Hop.

Kerberos Hop מתאר את האופן שבו אישורי Kerberos (כרטיסים) מועברים בין מערכות/שרתים, הכוונה היא: משתמש ניגש לשרת אחד (Hop ראשון), והשרת הזה צריך לגשת לשרת אחר בשמו של המשתמש (Hop שני).

זה נקרא delegation – השרת צריך להשתמש באישורי המשתמש כדי להמשיך הלאה.

**שלבי Kerberos Hop רגיל:**

התרחיש הוא כזה:

* + משתמש מתחבר מהמחשב שלו ל-ServerA (למשל דרך powershell remote)
  + ServerA מנסה לגשת לServerB (למשל לקרוא קובץ או להריץ שאילתה בשם המשתמש)

מה קורה בפועל:

1. המשתמש מקשבל TGT מהKDC
2. הוא מקבל גם Service ticket ל-ServerA
3. הוא ניגש לServerA ומזדהה
4. עכשיו – ServerA רוצה לגשת לServerB בשם המשתמש אבל אין לו את הTGT של המשתמש ולכן הוא לא יכול לבקש כרטיס נוסף.

ואז קורא הKerberos Double Hop Problem, ברירת המחדל של Kerberos לא מאפשרת Hop שני, כי כרטיס הTGT לא מועבר לServerA אלא רק הService Ticket לשרת זה, אלא אם יש delegation הHop השני ייכשל.

קורה כאשר משתמש מנסה לעבור את הcredentials שלו משני שרתים ברצף אבל כרטיס הכניסה (Kerberos ticket) לא עובר "hop" שני כברירת מחדל.

מתי זה קורה?

1. משתמש במחשב האישי שלו פותח חיבור (למשל RDP) לשרת A.
2. משרת A, הקוד או השירות מנסה לפנות הלאה לשרת B בשם המשתמש.
3. הפניה נכשלת – כי הcredentials של Kerberos לא מועברים אוטומטית מה-hop הראשון ל-hop השני.

למה זה קורה?

Kerberos נבנה כך ש:

* + TGT לא יועבר אוטומטית בין שרתים
  + זה נועד למנוע גניבת זהויות (credential theft) ע"י שרתים לא מהימנים.

מתי כן אפשר לעשות Hop? ולמה?

* + **CredSSP** - לדוגמא אפשר להתחבר מרחוק עם CredSSP בעזרת הפקודה:

Enter-PSSession -Authentication CredSSP

**איך זה עובד?**

המשתמש שולח את שם המשתמש והסיסמא שלו לServerA, ועכשיו ServerA מקבל גישה לכל האישורים (כולל סיסמא) והוא יכול להתחבר לServerB כי הוא מסוגל לבקש TGT מהKDC כי יש לו שם משתמש וסיסמא.

* + **Constrained Delegation** – כאשר המחשב ServerA מוגדר בAD לבצע delegation רק לשירותים ספציפים. אם בAD זה מוגדר אז הTGT לא נדרש, ServerA מקבל מהKDC גרטיס Service לServerB בשם המשתמש
  + **Protocol Transition + Constrained Delegation** – המשתמש התחבר עם פרוטוקול אחר (לדוגמא: NTLM) אבל רוצים להמשיך מול שרת שני עם Kerberos. זה עובד בכך זServerA מבצע impersonation של המשתמש (אפילו בלי שיש לו TGT), ואז מבקש KDC Service ticket לServerB. זה עובד מכיוון שאם Use any authentication protocol מסומן בAD עבור ServerA, זה מאפשר להתחיל Delegation.

1. **הסבר מהו Kerberos tolerance.**

הפער המרבי המותר בין השעון של הלקוח לשעון של הKDC שבו עדיין יאושר כרטיס Kerberos.

כלומר, אם השעון שלי והשעון של השרת לא מסונכרנים – כמה סטייה מותרת לפני שנחסם האימות.

ברירת המחדל היא 5 דקות, כלומר אם השעה במחשב הלקוח שונה מהשעה בDC ביוצר מחש דקות – האימות ייכשל

1. **הסבר: pass the hash, pass the ticket, silver ticket, golden ticket, diamond ticket, KERBEROAST.**
   1. **Pass the hash**

שיטה בה התוקף משתמש בhash של סיסמא (NTLM) במקום בסיסמא עצמה, כדי לאמת את עצמו מול מערכת Windows.

תנאים:

* + מצריך גישה מקומית או credentials dump
  + לרוב עובדים עם psexec או mimikatz
  1. **Pass the ticket**

התוקף משתמש בכרטיס Kerberos שכבר נוצר (TGT או TGS) כדי לגשת למשאבים. למשל: משתמש התחבר פעם אחת -> התוקף לוקח את ה-TGT מהמחשב שלו -> שולח אותו בעצמו לשרת אחר. לאחר מכן התוקף (בעזרת כלים כמו Mimikatz) טוען את הכרטיס בזכרון (LSASS) או לתוך Kerberos session של המערכת, פעולה זו גורמת לKerberos לחשוב שהתוקף הוא המשתמש המקורי. לאחר מכן התוקף יכול לעשות את כל מה שהמשתמש יכול לעשות מכיוון שיש לו את הticket של אותו משתמש.

* 1. **Silver ticket**

תוקף יוצר כרטיס מזוייף (TGS) ומתחזה למשתמש מול שירות (למשל SQL), הוא מנצל את הService Account כדי ליצור כרטיס TGS. כדי לייצר את הכרטיס התוקף צריך לדעת את של המשתמש והשירות (SPN) וגם גישה לסיסמא של המשתמש, בין אם זה הסיסמא עצמה (של הService account) או אם זה הNTLM hash. לאחר מכן התוקף יכול לייצר כרטיס TGS עם הנץונים הבאים:

* + שם משתמש שהוא רוצה לזייף (לרוב משתמש אדמין או אחר עם הרשאות)
  + ה-SPN של השירות
  + חתימה דיגיטלית עם המפתח הסודי של החשבון.

יתרון:

* + לא תלוי בDC (בניגוד לGolden ticket)
  + קשה לזיהוי
  1. **Golden ticket**

תוקף מזייף TGT בעזרת מפתח הזהב – מפתח הסימטרי של Kerberos (KRBTGT), התוקף יבצע Golden ticket אחרי שהוא גנב את המפתח של הkrbtgt (הNTLM hash שלו, כי אין לו סיסמא שמורה כמו שכבר גיליתי). זה בשלב מאוחר מכיוון שהמתקפה הזו מתבססת על הhash של הkrbtgt שהוא סודי ביותר בדומיין, אז צריך הרבה הכנות לפני שעושים את המתקפה.

תנאים:

* + הhash של החשבון KRBTGT שנמצא בDC
  + מאפשר גישה לכל שירות, בשם כל משתמש
  1. **Diamond ticket**

שילוב של יצירת TGT מותאם אישית + TGS שנרכש אמיתי. בגדול התוקף מדמה את עצמו למשתמש אמיתי, אבל רק לשירות ספציפי.

* 1. **KERBEROAST**

תוקף מבקש כרטיס שירות (TGS) עבור Service Account עם SPN, ומקבל אותו מוצפן ב-AES או RC4. ולאחר מכן התוקף מנסה לפענח את הכרטיס offline כדי למצוא את הסיסמא של הservice account.

תנאים:

* + המשתמש צריך להיות בתוך הדומיין
  + השירות צריך להיות מוגדר עם SPN

1. **איך מחשב מקבל כתובת ip אוטומטית?**

כאשר מחשב מוגדר לקבל כתובת IP באופן אוטומטי (DHCP), הוא מבצע תהליך שנקרה: DHCP DORA – שזה ראשי תיבות לארבעת השלבים בתהליך:

* 1. **Discover (D)**

המחשב משדר broadcast בקשה כדי לקבל IP, הבקשה כוללת:

* + כתובת MAC של הלקוח
  + בקשה לקבלת ip
  + אפשרויות נוספות כמו hostname, סוג שירות רצוי וכו
  1. **Offer (O)**

שרת הDHCP עונה עם הצעה: כתובת IP פנוייה + פרטים נוספים, ההצעה הזו גם היא נשלחת כbroadcast וכוללת:

* + כתובת IP שהשרת מציע ללקוח
  + פרמטרים נוספים: subnet mask, Default gateway, כתובות DNS, Lease time ועוד.
  + מזהה השרת Server identifier
  1. **Request (R)**

המחשב מבקש מהשרת לקבל את הכתובת שהוצעה לו, ההודעה כוללת:

* + כתובת הIP שהלקוח מבקש (זו שהשרת הציע)
  + מזהה השרת שמציע את הכתובת
  + מידע נוסף שנדרש לשרת כמו אפשרויות DHCP
  1. **Acknowledge (A)**

שרת הDHCP מאשר, ההודעה כוללת:

* + אישור הרשמי של כתובת הIP ללקוח
  + פרמטרים נוספים (כמו בשלב 2)
  + זמן השכירות (Lease Duration)

מה קורה:

1. מחברים מחשב לרשת
2. המחשב שולח DHCP Discover לכתובת 255.255.255.255
3. שרת הDHCP עונה ב- DHCP Offer עם כתובת לדוגמה: 192.168.1.25
4. המחשב מבצע DHCP Request
5. השרת עונה DHCP Ack – והכתובת נשמרת למחשב

פורטים:

* + UDP 67 – שרת DHCP מקבל.
  + UDP 68 – הלקוח מקבל.

במקרה והשרת לא זמין:

* + במחשב יחכה זמן מה
  + ואז יקבל כתובת זמנית אוטומטית בטווח APIPA (Automatic Private Internet Protocol Address) x.x.169.254 – שהמשמעות היא שאין רשת.

1. **מה הם directory partitions ומה ניתן לראות ב:**
   1. Schema
   2. Configuration
   3. Default naming context

מה זה?

Directory partitions הם חלוקות לוגיות של מסד הנתונים של הActive directory (קובץ NTDS.dit).

כל partition מכיל סוג אחר של מידע – ומסונכרן באופ שונה בין ה Domain controllers

כל DC מחזיק כמה מחיצות במקביל, שכל אחת משמשת מטרה אחרת.

שלושה סוגים מרכזיים:

* 1. **Schema**
  + קובע אילו סוגי אובייקטים קיימים בAD (User, Computer, Group, Printer וכו'')
  + מגדיר אילו תכונות יש לכל אובייקט (כמו mail)
  + נשמר רק בקבוצת ה Domain controllers שיש להם תפקיד FSMO (Flexible Single Master Operation) של Schema master
  + דוגמה: CN=Schema,CN=Configuration,DC=domain,DC=com
  + משותף לכל היער! (הסברתי בשאלות קודמות מה זה היער)
  1. **Configuration**
  + מכיל מידע על אתרים, קישורים בין אתרים, שירותים, רפליקציה, Certificate Authorities, ועוד.
  + לדוגמא: מיקומים של שרתי Global Catalog, פרטי שירותים מבוססים AD (כמו Exchange).
  + דוגמה: CN=Configuration,DC=domain,DC=com
  + משותף לכל היער!
  1. **Default naming context**
  + מידע האופרטיבי של הדומיין: משתמשים, קבוצות, מחשבים, יחידות ארגוניות (OU), GPOs.
  + משתנה מדומיין לדומיין
  + דוגמה: DC=company,DC=com
  + ייחודי לכל דומיין!

ניתן לראות את המחיצות האלו דרך כמה כלים, ביניהם:

* + Adsiedit.msc
  + Ldp.exe
  + powershell

1. **ברצוננו להוסיף לכל משתמש שדה חדש בAD בשם צבע שבו ירשם הצבע האהוב עליו, איך נעשה את זה?**

***חשוב: תהליך זה משפיע על כל היער! ולכן דורש זהירות יתרה והרשאות גבוהות מאוד. תהליך זה הינו בלתי הפיך***

בגדול, התהליך הוא כזה:

* + יוצרים תכונה חדש בשם שרוצים (נגיד FavoriteColor)
  + מקשרים אותה למחלקה של user
  + עכשיו ניתן לערוך את השדה הזה בפרופיל המשתמש

דרישות מקדימות:

* + נדרש שרת עם Schema Master
  + מחשב חבר בדומיין
  + משתמש חבר בקבוצת Schema Admins
  + גיבוי של ה-AD (מומלץ! לא חייב)
  + הפעלת קונסולת schema בעזרת: regsvr32 schmmgmt.dll

1. **קיימת רפליקציה בין שני DC. מחקנו משתמש בDC אחד. איך הDC השני יודע זה? ואיך קוראים למצב שבו המשתמש נכנס אליו?**

כל אובייקט בAD מתעדכן עם מספר גרסה (version) וספירת שינויים (USN – Update sequence Number).

כאשר נמחק משתמש מDC אחד, הוא לא נעלם מיד, במקום זאת מופיעה Tombstone – גרסה מיוחדת של האובייקט המסמלת שהוא נמחק.

Tombstone object:

* + אובייקט שנמחק לא נמחק מיידית – אלא מסומן כ-"deleted" ל180-60 יום (תלוי בהגדרה)
  + הDC שולח את המידע הזה לרפליקציה לשאר ה-DC
  + שאר הDC רואים את הTombstone ומוחקים גם אצלם את האובייקט.

כאשר משתמש מנסה להיכנס דרך DC שעדיין לא קיבל את המחיקה הוא מגיע למצב שנקרה "lingering object" (אפרט עליו שאלה הבאה)

1. **מהו lingering object?**

מצב כאשר אובייקט עדיין קיים בDC אחד, אך נמחק בDC אחר.

יכול לקרות כשDC היה מנותק מהרשת ולא קיבל את הTombstone בזמן.

אם משתמש כזה מנסה להתחבר – הוא יכול להתחבר דרך הDC הישן, מה שעלול ליצור בעיות אבטחה/עקביות.

(דרך אגב, המנגנון של הסנכרון עובד באותה הצורה, אם תרצי אסביר לך בדיוק איך)

יש לי סנכרון (ברשת סודי יש מסמך שמפרט בדיוק איך זה עובד) שלקח לי בערך 3 שנים לכתוב אותו בלי שהוא יישבר ויהיה כמו שצריך. אשמח להסביר לך אחרי זה איך כולו עובד אבל אסביר על הקטע שקשור לשאלה הזו:

רקע: זה סנכרון בין הSAP לDB מקומי על מחשב

הסנכרון פועל כל חמש דקות (ניתן להגדרה) מהצד של הDB המקומי, לא הסאפ.

כאשר מוחקים רשומה בסאפ, זה אמור להגיע לDB המקומי אבל המחשבים לא תמיד ברשת כדי לסנכרן, אז אני מסמן את הרשומה כ deleted עם שדה שהוספנו, ואז בסנכרון הבא הDB המקומי מקבל את השורה הזו כdeleted ומוחק אותה. ככה פתרתי את הבעיה של אם מחקתי אובייקט, והמחשבים לא היו ברשת, איך הם ידעו שזה נמחק?

האמת שלא ידעתי שיש לזה שם

הסברתי ממש בערפול כי לא רוצה להכנס כאן לפרטי פרטים, אבל זה מנגנון שעבדתי עליו רוב השירות (ביחד עם כל שאר הדברים), כל פעם שידרתי אותו שוב ושוב. הוא משלב windows service, cron job, ועוד כל מיני דברים מגניבים כאלה.

1. **לשם מה דרוש sites בad? איך מבוצעת חלוקה לsitesים?**

Sites משמשים לייצוג מבנה פיזי של הרשת – כלומר, סניפים/מיקומים גאוגרפיים שונים עם חיבורים שונים.

מטרות עיקריות:

* + אופטימיזציה של רפליקציה – שולט איך/מתי/כמה רפליקציה תתרחש בין DC
  + שיפור יעילות התחברות משתמשים – משתמשים "יודעים" לאיזה DC קרוב להתחבר
  + הפחתת תעבורה על קווים איטיים – אפשר להגדיר זמני רפליקציה שונים בין מיקומים
  + שליטה על GPOs מבוססי Site – אפשר להחיל מדיניות לפי מיקום פיזי
  + שירותים מבוססי מיקום – כמו Exchange, SCCM, DFS – משתמשים בSites לבחירת שרתים קרובים

איך מבוצעת החלוקה:

* 1. כל Site מייצג טווח IP פנימי (subnet) – כל מחשב שמתחבר לדומיין מזוהה לפי כתובת ה-IP שלו. לפי הטווח, הוא משויך לSite מסויים.
  2. ההגדרה מתבצעת ב-Sites and Services (dssite.msc)
     1. יוצרים Site חדש
     2. מקשרים אליו Subnets (למשל 192.168.10.0/24)
     3. מקשרים את ה-Domain Controllers שנמצאים פיזית בסניף הזה
     4. יוצרים Site Links – חיבורים בין האתרים עם משקלים (Cost) ואפשרויות רפליקציה (שעות, תדירות)

איך Sites משפיעים על רפליקציה?

* + רפליקציה בתוך אותו Site נחשבת "מהירה" – מתבצעת בתדירות גבוהה (ברירת מחדל: תוך שניות)
  + רפליקציה בין אתרים שונים נחשבת "איטית" – מתבצעת בדיליי קבוע (ברירת מחדל: 180 דקות) ונשלטת ע"י Site Link.

1. **איך מקימים trust בין שני דומיינים? מה זה מאפשר?**

\*\*יש את השלבים הכלליים של איך ליצור אמון דרך הGUI, אבל בגלל שאין לי את הGUI לא אוכל לפרט כמו שצריך וגם לא יודע אם מה שמצאתי תואם את הגרסאות האחרונות, לכן אראה רק את הדרך ליצור דרך הpowershell

**קשר מאובטח בין שני דומיינים מאפשר למשתמשים מאחד מהם לגשת למשאבים בדומיין השני (כפוף להרשאות).**

**דרישות מקדימות לפני הקמה:**

* + **שני הדומיינים פעילים ונגישים זה לזה ברשת (**ping, DNS**)**
  + **חיבור בין ה-**Domain Controllers
  + **הרשאות** Domain Admin **בשני הדומיינים**
  + DNS **שמוגדר כהלכה (כל דומיין צריך לדעת לפתור את שם הדומיין השני)**

**איך יוצרים?**

**אפשר דרך ה**powershell **בעזרת הפעולה:**

New-ADTrust -Name "domainB.com" -SourceForest "domainA.com" -Direction Bidirectional -TrustType External

**סוגי** Trust **שונים:**

* **Parent-Child – אוטומטי בתוך אותו** Forest
* **Tree-Root – אוטומטי בין עצים שונים באותו** Forest **– שני דומיינים שונים ץחץ אותו יער**
* **External Trust – ידני. בין דומיינים שונים ב**Forests **שונים – ארגונים נפרדים**
* **Forest Trust – ידני. בין שני** Forests **– ארגונים גדולים**
* **Shortcut Trust – לשיפור ביצועים בין דומיינים מרוחקים באותו היער – יער מורכב עם עומק**
* **Realm Trust – מול** Kerberos realm **לא-**AD **– אינטגרציה עם מערכות לא-מיקרוסופט**

**כיווניות:**

* **One-Way – דומיין** A **סומך על** B**, אבל** B **לא סומך על** A**, משתמשים בB יכולים לגשת למשאבים בדומיין A**
* **Two-Way – כל אחד סומך על השני.**

1. **ישנו משתמש שננעל כל 15 דקות. איך אנחנו יכולים לדעת מאיפה הוא ננעל?**

לרוב, משתמש ננעל כאשר יש Job, שירות, מחשב או התקן אחר שמנסה לבצע התחברות עם סיסמה ישנה או שגויה.

כדי למצוא מאיפה הוא ננעל, נשתמש בשיטות אבחון שונות, כמו Event Log, Powershell וכלים של מייקרוסופט

שלבים לאיתור מקור הנעילה:

* 1. **למצוא את הDC שמבצע את הנעילה** – ניגש לDomain controller שמרכז את הנעילה, ונחפש בEvent Viewer את האירועים הבאים:  
     **4740** – חשבון ננעל  
     **4625** – ניסיון התחברות נכשל  
     ישנו גם שדה חשוב באירוע **4740** שנקרא: Caller Computer Name שזהו שם המחשב שממנו הגיע הבקשה שהביאה לנעילה
  2. **לעקוב אחרי השרת או המחשב שהוביל לנעילה** – ברגע שיש לנו את שם המחשב אנחנו צריכים:
  + להתחבר אליו מרחוק או לבקש מהמשתמש להפעיל אותו
  + לבדוק אם יש שירות שמוגדר לרוץ כמשתמש זה
  + לבדוק אם יש Mapped Drives או Scheduled Tasks
  + אם יש Session ישן פתוח
  + אם יש Credential manager עם הרשאות שמורות לא נכונות
  1. **אפשר להשתמש בכלי אבחון של מייקרוסופט, לא אסביר יותר מידי אבל אלא חלק מהכלים האפשריים:**
  + LocoutStatus.exe – מציג את הסטטוס של המשתמש בכל הDCs
  + ALTools – כלים נוספים לבחינת נעילות
  1. **Powershell חיפוש נעילות בזמן אמת** – אפשר לרשום קוד powershell שמחזיר לנו את המידע שאנחנו מבקשים, הנה הסקריפט:

Get-EventLog -LogName Security -InstanceId 4740 -Newest 1000 |

Where-Object { $\_.Message -like "\*username\*" } |

Select TimeGenerated, MachineName, Message

1. **נמחק חשבון של משתמש מהad, איך אפשר לראות מי המשתמש שמחק אותו?**

אותו דבר כמו השאלה למעלה, יש פעולה שנרשמת ל Event Log וניתן למצוא אותה דרך הEvent Viewer.

**4726** – חשבון נמחק

אפשר למצוא בתוכן של אירוע זה את הנתונים הבאים:

* + Subject – המשתמש שביצע את המחיקה
  + Target Account – שם החשבון שנמחק
  + Caller Computer – מחשב ממנו בוצעה הפעולה

כמובן, שכמו השאלה למעלה, אפשר לכתוב קודי powershell או משהו בסגנון כדי לקבל את כל המידע שאנחנו רוצים. (עשיתי גם כמה כאלה)

1. **איך בוחרים איזה מידע ירשם לeventlog של הDC?**

אפשר על ידי הגדרה של Audit Policies שמוגדרת לרוב באמצעות Group Policy (GPO).

[לא הבנתי, לברר]

1. **איך נמצא את כל המחשבים ממערכת הפעלה מסויימת (לא powershell)?**

ניתן לעשות זאת דרך ADUC (Active Directory Users and Computers) בעזרת שאילתה מותאמת.

ניתן להריץ את הADUC בעזרת הפעולה dsa.msc.

ומשם להריץ את הפעולה דרך הGUI.

\*\*\*שוב, לא חושב שהכרחי שאגיד איפה ללחוץ, אפשר למצוא את זה בקלות באינטרנט.

\*\*\*אני מאוד אוהב שאילתות ודברים כאלה, ופחות אוהב GUI, סתם שתדעי, כי בGUI הרבה יותר קשה (אבל אפשרי) לעשות קודים אוטומטים או קיצורי דרך, חוץ מזה, GUI יכול להשתנות, השאילתות, פחות כי זה איך זה עובד מאחורה.

דוגמא לשאילתת LDAP - (&(objectCategory=computer)(operatingSystem=Windows 10\*))

1. **מדוע קיימת ההרשאה change password לeveryone על חשבונות מחשב?**

כדי לאפשר למשתמשים לשנות את הסיסמא של עצמם גם אם אין להם את ההרשאה המתאימה באובייקט AD.

ההרשאה הזו לא מאפשרת לשנות סיסמא לאנשים אחרים, רק לעצמי.

מי שיכול לשנות את הסיסמא שלי הוא או מנהל עם ההרשאה “Reset Password” (מנהל) או אני עצמי במחשב שלי.

ההרשאה הזו מאפשרת לכל ישוב ברשת (כולל משתמשים ומחשבים) לנסות לשנות את סיסמת החשבון המדובר, כאן הכוונה היא לשינוי סיסמה של חשבון המחשב בעצמו.

למה זה קיים לEveryone?

מחשבים בAD מחליפים את סיסמת החשבון שלהם אוטומטית כל תקופה (ברירת מחדל: 30 יום), כדי לבצע את השינוי הזה, המחשב צריך את היכולת לשנות את סיסמת החשבון שלו בעצמו, מבלי צורך בהרשאות גבוהות, לכן ההרשאה הזו ניתנת ל-Everyone כדי שהמחשב עצמו, באמצעות שיאותי הרשת, יוכל לשנות את הסיסמה שלו.

צורה זו לא מאפשרת לכל אחד "לא מאומת" לשנות את הסיסמה של כל מחשב מכיוון שההרשאה היא על האובייקט עצמו (החשבון של המחשב) אבל רק הישות שמזוהה כ"בעל החשבון" (המשתמש או המחשב עצמו) יכול לבצע אצ השינוי בהצלחה.

זה אומר שרק המחשב או המשתמש שמזוהה כבעל החשבון יכול לשלוח בקשת שינוי סיסמה שמאומתת באמצעות NTLM/Kerberos. אם ישות אחרת תנסה, הבקשה תידחה.

למה זה הכרחי?

החלפת הסיסמה האוטומטית של מחשבים מונע מצב של סיסמאות סטטיות שנשארות לנצח, מה שמשפר את האבטחה. בלי הרשאה זו לEveryone, המחשבים לא יוכלו לעדכן את סיסמת החשבון שלהם בלי התערבות של מנהל, וזה לא יהיה כיף.

1. **מה זה GPO? איך אפשר לנצל אותו לרעה?**

זה מנגנון בActive Directory שמאפשר לנהל ולהחיל הגדרות בצורה מרוכזת על משתמשים ומחשבים בדומיין.

ה-GPO שולט על:

* + הגדרות מערכת והפעלה
  + התקנת תוכנות
  + הרשאות קבצים ורג'יסטרי
  + סקריפטים להפעלה/כיבוי
  + הגבלות אבטחה והרשאות משתמשים.

איך זה עובד?

GPO מקושר ל Sites / Domains / Organizational Units (Ous)

מוחל על אובייקטים (משתמשים או מחשבים) שנמצאים תחת האובייקט אליו הוא מקושר.

ניתן לנהל GPO דרך הכלי Group Policy Management Console (gpmc.msc)

איך אפשר לנצל לרעה?

כאשר תוקף משתלט על סביבת הAD, שימוש בGPO היא אחת הדרכים החזקות ביותר לשליטה.

* + אפשר להריץ קוד זדוני דרך Stratup/Logon scripts
  + אפשר להפיץ תוכנות דרך ה Software Installation
  + אפשר להגיד Scheduled Tasks דרך הGPO – המשימות האלה יכולות לרוץ עם הרשאות של מערכת, או לקחת מידע על מחשב של משתמשים ולשלוח אותו לאינשהו מרכזי
  + להזריק DLL דרך AppInit\_DLLs – אפשר לקבוע רג'יסטרי שיטען DLL זדוני בכל אפליקציה שמשתמשת בUser32.dll
  + השגת Persistency דרך הרשאות וגישה – הפיכת משתמשים רגילים למנהלים מקומיים על כל מחשב בארגון, ובכך אפשר להשתלט על כל מחשב, לא רק של מנהל.
  + שינוי סיסמאות או יצירת משתמשים חדשים

בגדול, אפשר לעשות הרבה בלאגן.

1. **מהם השיתופים** [**\\domain.com\sysvol**](file:///\\domain.com\sysvol)[**\\domain.com\netlogon**](file:///\\domain.com\netlogon)

שני שיתופים קריטיים שנוצרים אוטומטית על Domain Controllers בכל סביבה של Active Directory.

[\\domain.com\SYSVOL](file:///\\domain.com\SYSVOL)

* + תיקייה שמכילה את קבצי המדיניות הקבוצתית (GPO) ואת הסקריפטים של Logon/Startup/Shutdown של משתמשים ומחשבים.
  + מסונכרנת בין כל ה-Domain Controllers בארגון באמצעות רפליקציה (DFS-R)
  + נתיב מקומי: C:\Windows\SYSVOL\sysvol\<domain>\Policies\

שימושים:

* + החלת GPO על מחשבים ומשתמשים
  + אחסון סקריפטים שירוצו עם התחברות או הפעלה
  + עותק קריא בלבד למחשבים בדומיין

לכל המחשבים בדומיין יש גישה לשם מכיוון שזהו המנגנון שאחראי על הפצת GPO וסקיפטים למחשבים ולכן הוא חייב להיות נגיש.

איך זה עובד?

כל מחשב מחובר לדומיין יודע את שם הדומיין, לדוגמה: some.domain.

כאשר אני ניגש ל [\\some.domain\SYSVOL](\\\\some.domain\\SYSVOL), זה שם DNS מיוחד שמפנה אותי לאחד משרתי הDC. לאחר מכן המחשב פותח חיבור SMB (שיתוף קבצים של ווידוס) ומוריד את הסקריפטים או ההגדות שצריך לפי הGPO.

[\\domain.com\NETLOGON](file:///\\domain.com\NETLOGON)

* + תיקיית שיתוף המיועדת בעיקר לסקריפטים להפעלה והרשמה (Logon scripts) וגם לקבצי תצורה ישנים.
  + שימשה ביסטורית בעיקר בWindows NT / 2000 אבל עדיין קיימת ונמצאת בשימוש גם היום
  + גם היא לקריאה בלבד עבור משתמשים רגילים
  + נתיב מקומי: C:\Windows\SYSVOL\sysvol\<domain>\scripts\

שימושים:

* + סקריפטים של התחברות (logon.bat, logon.ps1 וכו)
  + הפצת קבצי תצורה קטנים (Configuration files)
  + גישה לשיתופים כלליים מה-GPO

1. **מה אפשר לעשות עם הפקודה dsacls? למה נשתמש בה?**

כלי בCMD של windows שמאפשר להציג ולשנות ברשאות (ACL) על אובייקטים בActive Directory.

הפקודה מאפשרת:

* 1. להציג הרשאות על אובייקט ב-AD (למשל משתמש, OU, קבוצות)
  2. להוסיף, להסיר או לשנות הרשאות על האובייקט
  3. לבדוק מי יכול לעשות מה – כולל Delegation והרשאות נסתרות

שימושים נפוצים:

* + בדיקת הרשאות של משתמש
  + בדיקת Delegation
  + בדיקת חולשות אבטחה – תוקף יכול להשתמש בזה כדי למצוא אובייקטים שהוא יכול לשלוט בהם
  + שינוי הרשאות

פורמט הרשאות:

R – קריאה בלבד

RP – Read property

WP – Write Property

CA – Control Access

GA – Generic All

RC – Read Control

WD – Delete

תוקף יכול להשתמש בזה כדי למצוא אובייקטים עם הרשאות שגויות, לדוגמא משתמש רכיל שיש לו הרשאה למחוק משתמש אחר או לשנות סיסמא למשתמשים אחרים.

1. **איזה עוד פקודות dsXXX קיימות ומה ניתן לעשות איתם?**
   * Dsquery – מחפש אובייקטים בAD
   * Dsadd – יוצר אובייקטים חדשים בAD
   * Dsmod – משנה תכונות של אובייקט קיים
   * Dsrm – מוחק אובייקטים בAD
   * Dsmove – מעביר אובייקטים ממיקום אחד לאחר (למשל OU)
   * Dsget – מציג מידע על אובייקט
   * Dsacls – מציג או משנה הרשאות (כמו שרשמתי למעלה)
2. **איך אפשר לראות את הטיקטים של הקרברוס שלנו?**

יש פקודה כדי לראות את הטיקטים שהמחשב שלי מחזיק – כלומר הכרטיסים שמונפקים למשתמש הנוכחי על ידי הKDC (Key Distribution Center).

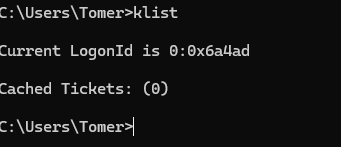
אתייחס לפקודה של ווידוס.

Klist

הפקודה תציג:

* + TGT – כרטיס כניסה ראשוני
  + Service Tickets – כרטיסים לשירותים (למשל MSSQL, גישה לשרת קבצים וכו)
  + תאריך תפוגה
  + Realm / domain
  + סוג הצפנה
  + **אם מריצים את הפקודה כמנהל אפשר גם לראות את הכרטיסים של משתמשים אחרים (באותו המחשב מן הסתם)**

ניסיתי לעשות את זה אצלי, זה מה שיצא כי אני לא התחברתי דרך Kerberos



אבל הנה דוגמא שמצאתי באינטרנט:

Current LogonId is 0:0x5a2e

Cached Tickets: (2)

#0> Client: user@DOMAIN.COM

Server: krbtgt/DOMAIN.COM@DOMAIN.COM

KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96

Start Time: 6/17/2025 09:01:00

End Time: 6/17/2025 18:01:00

Renew Time: 6/24/2025 09:01:00

1. **מה זה PKI. (לא המפלגה הקומוניסטית הפקיסטנית) איך זה בא לידי ביטוי אצלנו ברשת?**

Public Key Infrastructure – מערך טכנולוגי לניהול אימות וזהות דיגיטלית האמצעות הצפנה אסימטרית, זה הבסיס למרבית האבטחה באינטרנט, בארגונים ובמערכות מידע מודרניות.

הצפנה אסימטרית היא הצפנה שמשתמשת בזוג מפתחות: מפתח ציבורי ומפתח פרטי.

זה דומה להצפנה סימטרית שבה משתמשים באותו המפתח גם כדי להצפין וגם כדי לפענח, באסימטרי מצפינים עם המפתח הציבורי ומפענחים עם המפתח הסודי.

Certificate, זה בתכלס תעודה שבא אני אומר "אני תומר מור האמיתי" כי ברשת, את לא יכולה לסמוך על אף אחד, כל אחד יכול להגיד שהוא תומר מור, אז פה נכנס הCertificate, שזה בעצם מסמך דיגיטלי שמזהה באופן ייחודי ישות (כמו אדם, שרת או ארגון) ומוכיח את הזהות שלה בצורה אמינה ומאובטחת.

מה הוא כולל?

* + **תעודת זהות דיגיטלית** **(Certificate) –** מזהה אדם/שרת/מכשיר בצורה מוצפנת
  + **מפתח ציבורי** – משמש להצפנה ואימות
  + **מפתח פרטי** – נשמר בסוד, משמש לפענוח וחתימה
  + **CA –** **Certificate Authority** – הגורם שמנפיק תעודות ומאשר זהויות
  + **CRL/OCSP** – מנגנונים לבדוק אם תעודה בוטלה או פגה
  + **RA – Registration Authority** – גוף שאוסף ומוודא מידע לפני הנפקת תעודה.

איך זה עובד?

1. משתמש או שרת מבקש תעודה
2. RA מזהה אותו (למשל לפי שם משתמש, IP, אימייל)
3. CA מנפיק תעודת X.509 עם המפתח הציבורי של המבקש
4. הצדדים יכולים להשתמש בתעודה כדי: להצפין מידע, לוודא שהזהות אמיתי, לוודא שהמידע לא שונה.

[חסר לי, קשור לרשת שלנו]

1. **כיצד ניתן לכבות את ה audit של מחשב ב security log.**

פעולה רגישה מאוד, היא פוגעת ביכולת לנטר פעולות. זו גם טכניקה שתוקפים משתמשים כדי להסתיר עקבות!

* 1. **באמצעות Group Policy (דרך ממשק רגיל)**

אפשר לפתוח gpedit.msc (אם זה מחשב אישי) או gpmc.msc (אם זה בAD)

* 1. **באמצעות auditpol.exe (CMD)**

דוגמא לסקריפט שמבטל הודעות הצלחה וכישלון של logon

auditpol /set /subcategory:"Logon" /success:disable /failure:disable

לאחר הפעולה הזו, לא יווצרו לוגים שקשורים להתחברות, גם הצלחה וגם כישלון (מספרים: 4624,4625,4647,4634). הפעולה עצמה עלולה ליצור לוג אבל רק אם מופעלת ביקורת לשינויים במדיניות. ואם כן, האירוע שצפוי להיווצר הוא 4719, System audit policy changed

* 1. **מחיקת או השבתת Security Log**

יש כלים כמו wevtutil או powershell כדי למחוק Security logs.

* 1. **משנים הרשאות לקובץ**

C:\Windows\System32\winevt\Logs\Security.evtx

כדי למנוע כתיבה ללוגים.

1. **מהם windows Security principals ומה ההרשאות של כל אחד מהם?**

מתייחס לכל ישות (entity) שיכולה לקבל הרשאות ולהיכלל ב-ACL. זה כולל משתמשים, קבוצות, מחשבים ועוד.

Security principal – כל אובייקט שיכול להיות מזוהה במערכת מקבל Security Identifier

סוגים שונים:

* 1. **User** – משתמש בודד, SID ייחודי, הרשאות מותאמות לפי תפקיד
  2. **Group** – קבוצה של משתמשים, SID קבוע או מותאם, מקבלת הרשאות בשם כל חבריה
  3. **Computer** – כל מחשב חבר בדומיין, **S-1-5….**, גישה לשירותים או סקריפטים
  4. **Everyone** – כל משתמש, גם לא מזוהה, **S-1-1-0**, כברירת מחדל, הרשאות בסיסיות בלבד
  5. **Authenticated Users** – כל משתמש שעבר אימות, **S-1-5-11**, גישה מועדפת על פני Everyone
  6. **Anonymous Logon** – משתמש לא מזוהה, **S-1-5-7**, כמעט ולא ניתנות לו הרשאות
  7. **Local system** – מערכת ההפעלה עצמה, **S-1-5-18**, גישה מלאה לכל דבר
  8. **Network Service** – פועל עם הרשאות מוגבלות ברשת, **S-1-5-20**, גישה מוגבלת אך רשתית
  9. **Local Service** – שירותים עם גישה מוגבלת למערכת, **S-1-5-19**, הרצת שירותים ללא הרשאות מנהל
  10. **Administrators** – קבוצת מנהלים מקומית, **S-1-5-32-544**, גישה מלאה
  11. **SYSTEM** – החשבון הפנימי של Windows, כמו LocalSystem, שליטה מוחלטת בכל

מתי רואים אותם?

* + בהגדרת ACL לקובץ או תיקייה (Security tab)
  + בקביעת הרשאות בGPO
  + כשבודקים לוגים באבטחה – מופיעים הSID או שם משתמש/קבוצה

1. **מה זה LDAP?**

עשיתי בעבודת בית.

1. **רשום שאילתת LDAP המחפשת את כל המשתמשים שהמספר האישי שלהם מתחיל ב-60.**

\*\*רשמתי גם שאילתות כאלה בשירות, אשמח לספר לך אם תרצי, האמת רשמתי שאילתה כזו בדיוק!

נניח שהשדה ששומר את המידע הזה נקרא employeeID, זו השאילתה המתאימה:

(&(objectClass=user)(employeeID=60\*))

ה & זה בגלל שאנחנו משתמשים בשני סינונים בו זמנית, אנחנו מחפשים על כל האובייקטים מסוג של user שה employeeID מתחיל ב60, ה\* זה wildcard מה שאומר, כל דבר.

**אני יודע בדיוק מה השדה הספציפי אבל אני גם אחראי אז לא רציתי לרשום אותו. (עד שמצאתי אותו יצאה לי הנשמה)**

1. **מה זה LAPS?**

Local Administrator Password Solution

פתרון אבטחה של מיקרוסופט שמנהל באופן אוטומטי וסודי את הסיסמא של חשבון ה-Administrator במקומי על כל מחשב בארגון.

למה צריך את זה?

ברירת המחדל של הרבה ארגונים היא שלכל המחשבים יש את אותה סיסמה לAdministrator מקומי, וזה סיכון גדול.

מה הוא עושה?

* 1. מגדיר סיסמא ייחודית (חזקה ואקראית) לחשבון אדמין על כל מחשב מקומי
  2. מאחסן את הסיסמא בActive Directory בתוך attribute של אובייקט המחשב.
  3. הAD מגביל את מי שרואה את הסיסמא לפי הרשאות
  4. מרענן את הסיסמא האוטומטית בפרק זמן קבוע (למשל 30 יום)

רכיבים עיקריים:

1. GPO – מגדיר איך עובד, משך זמן בין חלופות וכו
2. LAPS Client – מוצקן על תחנת קצה (מחשב אישי), יוצר ושולח סיסמאות לAD
3. Attribute ב AD – כדי לשמור את הסיסמא שנוצרה
4. **איך psexec עובד?**

מאפשר להריץ פקודות מרחוק על מחשבים אחרים ברשת – כאילו עובדים מולם פיזית.

איך זה עובד?

PsExec משתמש בפרוטוקול SMB (פורטים 455 / 139), כדי לתקשר עם המחשב המרוחק.

כאשר מריציפ פקודה עם PsExec הדברים הבאים קורים:

1. הכלי מעלה למחשב היעד קובץ בשם PSEXESVC.EXE לתיקייה [\\target\ADMIN$](file:///\\\\target\\ADMIN$) (לרוב C:\Windows)
2. הוא יוצר שירות זמני בשם psexesvc ומריץ אותו
3. השירות מריץ את הפקודה הרצויה עם ההרשאות שסופקו
4. לאחר סיום – השירות מוסר, והקובץ נמחק.

הפקודה רצה על המחשב המרוחק והפלט שלה חוזר למחשב המקומי, כמו ipconfig לדוגמא:

PsExec.exe \\PC01 -u domain\admin -p MyPassword ipconfig

[\\PC01](file:///\\PC01) – שם המחשב המרוחק או כתובת IP

-u, -p – פרטי התחברות (שם משתמש וסיסמא)

Ipconfig – הפקודה שתרוץ

דרישות להפעלה:

* + **הרשאות אדמין** – על המחשב המרוחק המשתמש חייב להיות אדמין
  + **ADMIN$ עם שיתוף פעיל** – משתמש להעלאת הקובץ המרוחק
  + **פתיחת פורטים** – SMB בעיקר 445
  + **Firewall** – חייבת לאפשר גישה מהסוג הזה
  + **Credentials Delegation** – לפעמים דורש הגדרות SPN/Delegation לפעולות מורכבות יותר

1. **מה זה DFS?**

Distributed File System

שירות של מיקרוסופט שמאפשר ליצור ולהפיץ שיתופי קבצים ממיקומים שונים ל-מרחב קבצים ורטואלי אחד (Share), ולהפוך את חווית המשתמש לשקופה, עקבית ונגישה – בלי שיידע איפה הקבצים באמת מאוחסנים

מאפשר ליצור כתובת רשת אחת קבועה (כמו [\\domain.com\FIles](file:///\\\\domain.com\\FIles)) שיכולה להצביע לקבצים מאוחסנים בשרתים שונים ומפוזרים – בארץ, בחול, לפי מחלקות או מיקומים גאוגרפיים

סוגי DFS:

1. **DFS Namespace** – יוצר עץ תיקיות וירטאלי שנראה כמו Share רגיל, המשתמש לא רואה את המיקום הפיזי של הקובץ, רק את הנתיב הלוגי (לדוגמא: [\\domain.com\Files\Somefile.txt](file:///\\domain.com\Files\Somefile.txt)) ומנתב אוטומטית לשרת רלוונטי
2. **DFS Replication (DFSR)** – מסנכרון תיקיות בין שרתים שונים, משתמש ב Remote Differential Compression (RDC) כדי לשלוח רק את השינויים, אפשר לקבוע תזמון, כיווניות (דו כיווני או חד כיווני) והעדפות.
3. **מה זה SCCM?**

System Center Configuration Manager

זה פתרון ניהול מרכזי של תחנות קצה מבית מייקרוסופט.

הוא מאפשר לארגונים לנהל, לפרוס, לעדכן ולאכוף מידיניות על אלפי מחשבים, שרתים, לפטופים ואפילו מכשירים ניידים – ממקום מרכזי אחד

מה הוא יודע לעשות?

* + **הפצת תוכנה** – התקנת תוכנותף עדכונים, סקריפטים,חבילות
  + **ניהול עדכונים (WSUS)** – הורדה והתקנת עדכוני Windows ואבטחה
  + **ניהול הגדרות (Compliance)** – אכיפת הגדרות מדיניות (דפדפן, חומת אש וכו)
  + **ניטור ודיווח** – מידע על חומרה, תוכנה, עמידה בתקנים
  + **OS Deployment (OSD)** – פריסת מערכות הפעלה אוטומטית
  + **Remote Control** – שליטה מרחוק על מחשבים
  + **אינטגרציה עם intune** – ניהול מכשירים בענן ובמקומי

רכיבי המערכת:

* + **Site Server** – השרת הראשי שמנהל את הכל
  + **Distribution Point (DP)** – שרת שמחזיק חבילות להתקנה, עדכונים, ISO וכו
  + **Management Point (MP)** – נקודת תקשורת בין לקוחות לשרת SCCM
  + **Client Agent** – תוכנה שמותקנת על כל תחנת קצה ומבצעת פקודות מהשרת
  + **Software Update Point (SUP)** – שרת שמספק עדכוני WSUS ל SCCM

1. **מה זה Credentials Guard? איך זה עובד? למה זה מקשה עלינו לעשות מתקפות מבוססות hash-ים?**

מנגנון אבטחה של מיקרוסופט שמטרתו למנוע גניבת זהויות, למשל כמו Pass-The-Hash או Pass-The-Ticket.

מה זה?

זה פיצ'ר שמפריד את האחסון של הרשאות רגישות (כמו hashים של סיסמאות וTGT של Kerberos) מהמערכת הראשית על ידי הרצת רכיבי LSASS בסביבה מבודדת (VM).

הוא מבוסס על VBS (Virtualization-Based Security) , שמוסבר שתי שאלות אחרי זאת, ועובד רק בWindows 10/11 Enterprise or Education.

איך זה עובד?

1. מערכת ווידוס מריצה LSASS (Local Security Authority Subsystem), זהו שירות רגיל אשר אחראי על ניהול הרשאות, סשנים, hash-ים, tickets וכו)
2. כאשר ה Credential Guard מופעל:
   1. חלקים רגישים ב LSASS (כמו סיסמאות וhashים) עוברים לאחסון בתוך סביבה וירטואלית מוגנת (Isolated LSA)
   2. מערכת ההפעלה הרגילה לא יכולה לקרו את המידע הזה – גם לא תוקף עם SYSTEM או עם Mimikatz
3. רק LSASS (המבודד) יכול לגשת לcredentials, וכל שאר הרכיבים (כולל תוכנות זדוניות) חסומים.

למה זה מקשה עלינו?

* + **Pass the hash** – לא ניתן לשלוף hash של סיסמא מתוך הזיכרון של LSASS כי הוא לא שם.
  + **Mimikatz** – לא מצליח להוצא hash או TGT/TGS כשהמערכת מוגנת
  + **Pass the ticket** – אין גישה ל Kerberos tickets, הם מוגנים בזכרון מבודד.
  + **Credential Dumping** – נחסם כי הcredentials לא נמצאים בLSASS הרגיל אלא באיזור מבודד.

1. **מה זה UAC? איך זה עובד? איך אפשר לעקוף אותו?**

עשיתי בעבודת בית.

1. **הסבר על Virtualization based security**

מנגנון אבטחה מתקדם שקיים במערכות הפעלה של מיקרוסופט (בעיקר בגרסאות windows 10 ואילך), המנגנון משתמש ביכולות וירטאליזציה של החומרה כדי ליצור סביבה מבודדת ומוגנת בזיכרון שבה ניתן להריץ שירותי אבטחה רגישים.

איך זה עובד?

במקום להריץ את כל רכיבי מערכת ההפעלה באותה סביבת kernel, VBS משתמש בטכנולוגיית Hyper-V כדי ליצור "אזור מאובטח" בזכרון.

רכיבים עיקריים:

* + **LSA Protection** – מבודד את lsass.exe כיד להגן על סיסמאות וכו
  + **Credential guard** – מוסבר למעלה
  + **HVCI (Hypervisor-Enforced Code Integrity)** – מבטיח שרק קוד חתום ומאומת ירוץ בזיכרון הקרנל.
  + **Secure Kernel** – ליבה מבודדת שפועלת במקביל ל Windows kernel אך נמצאת תחת בקרה של Hypervisor

1. **השגת shell על עמדה שמוחבר אליה Domain admin. איך אתה יכול להשתלט על המשתמש שלו? הסבר כמה דרכים והסבר על היתרונות והחסרונות של כל אחת.**

כמו שרשמתי בכמה מהתשובות המבחן הזה אפשר להשתמש בטכניקות כמו, לא אסביר עליהם יותר מידי:

**Pass the hash**

לקחת את ה hash של המשתמש ולהשתמש בו, בלי צורך בסיסמא המקורית.

יתרונות:

* לא מצריך את הסיסמא עצמה
* יעיל מאוד בסביבות שמשתמשות בNTLM

חסרונות:

* מוגבל לNTLM, לא עובד אם נדרש Kerberos
* Credential Guard מגן על הHashים

**Pass the ticket**

התוקף משתמש בכרטיס Kerberos שכבר נוצר (TGT או TGS) כדי לגשת למשאבים.

יתרונות:

* שימוש בזהות המלאה של המשתמש עם Kerberos
* קל מאוד כשיש session פעיל של DA

חסרונות:

* הכרטיסים פגים (פג תוקף) לאחר זמן (לרוב 10 שעות)
* דורש גישת SYSTEM כדי להוציא אותם.

אפשר גם להשתמש בעוד מתקפות כמו Golden ticket, Token Impersonation ו Mimikatz

1. **נניח שנחתנו על עמדה שיש לנו הרשאות אדמין מקומי עליה ומחובר אליה משתמש שאנחנו מעוניינים בהרשאות הדומייניות שלו, אבל לא הצלחנו להוציא את ה-hash בגלל שיש credential guard על העמדה. מה היית עושה כדי להצליח להשתמש בהרשאות של המשתמש שמחובר לעמדה (למשל כדי לגשת למיקום מסויים בנטאפ או לאפס סיסמא למשתמש אחר)**

אופציה מפגרת (קרדיט לאח שלי): אפשר להתקין/לשים KeyLogger על מחשב ואז לנעול אותו בכוח כדי שהוא יכניס את הסיסמא מחדש ואז Vualla יש לנו את הסיסמה של המשתמש עם ההרשאות הדומנייניות, איזה כיף!

אפשר, במקום לנסות לגנות לו את ההרשאות, לרכב על החיבור הקיים עם פקודה כמו:

net use \\netapp.domain.local\share /user:DOMAIN\user

1. **מה זה ADCS? פרט על טכניקת הרשאות הקשורה לcertificates**

Active Directory Certificate Services

רכיב של Windows Server שמספק שירותי PKI בסביבת Active Directory. המערכת מאפשרת הנפקה, ניהול וביטול של תעודות דיגיטליות לשם:

* אימות
* חתימה דיגיטלית
* הצפנת מידע
* אימות מכשירים או משתמשים

מבנה בסיסי:

1. **Root CA** – הסמכות העליונה שמחתימה את ה Sub Cas
2. **Enterprise CA** – CA שמשולב עם Active Directory, מאפשר הנפקת תעודות אוטומטית למשתמשים ומחשבים.
3. **Certificate Templates** – תבניות שמגדירות איך נראת תעודה, מי יכול לבקש אותה, ולמה היא משמשת.
4. **Enrollment Services** – שירות שדרכו משתמשים או מכשירים מבקשים תעודות
5. **CRL / OCSP** – רשימות ביטול תעודות או שרתי בדיקה אונליין.

טכניקת ברשאות הקשורה לCertificate:

ESC3 – שליטה על שרת הCA עצמו.

יכול להנפיק כל תעודה שהוא רוצה, ולזייף זהויות.

\*\* לא בטוח שהבנתי את החלק השני, אבל זה מה שמצאתי, כולל עוד כמה טכניקות כמו ESC1, ESC2 וכו

1. **הסבר על AD shadow credentials**

טכניקת תקיפה מתקדמת ב Active Directory בה התוקף משיג גישה מתמשכת על ידי רישום Credential חדש סמוי (צל) לחשבון משתמש או מחשב – מבלי לשנות את הסיסמא שלו או להצטרך לסיסמה המקורית.

**איך זה עובד?**

1. **התוקף מקבל הרשאות לשנות את ה**attributes **של משתמש מסוים ב**AD **– למשל דרך** Delegation **או השתלטות.**
2. **הוא מייצר זוג מפתחות (**private/public**)**
3. **את ה** public **הוא מוסיף ל** msDS-KeyCredentialLink **של המשתמש**
4. **באמצעות הכלי** PKINIT **התוקף יכול לקבל** TGT **כהמשתמש הזה – גם בלי סיסמא ומבלי לגרום לשינוי בסיסמא.**
5. **למי יש דומיין אדמין?**

**אני בטוח שזאת לא התשובה אבל:**

**לאדמין ולהאקר שפרץ למערכת.**