digital whisper
Technologic Papers

Kerberos Delegation 101

מאת ספיר פדרובסקי

הקדמה

Kerberos הינו פרוטוקול המאפשר אימות מאובטח בסביבת Windows Domain (ליתר דיוק, מאובטח יותר לעומת קודמו NTLM, שלצורך פירוט הפערים האבטחתיים בו נצטרך מאמר נוסף. עם זאת, יש מספר מאמרים מעולים על NTLM, הפגיעויות שלו וההבדלים מול Kerberos, אצרף קישורים בסוף המאמר).

ככל שנברתי בפרוטוקול, הבנתי שהעומק שלו, הפיצ'רים שלו והאופציות לנצל אותו לצורך הסלמת הרשאות בדומיין הם עולם ומלואו. במאמר זה, אנסה להציג בקצרה פיצ'ר מאוד מעניין של הפרוטוקול, שגם בתוכו, שיטות הניצול השונות הולכות ומתפתחות בקצב מסחרר. כמו כן, אציג מספר מתקפות בנושא ומקרי קצה מעניינים.

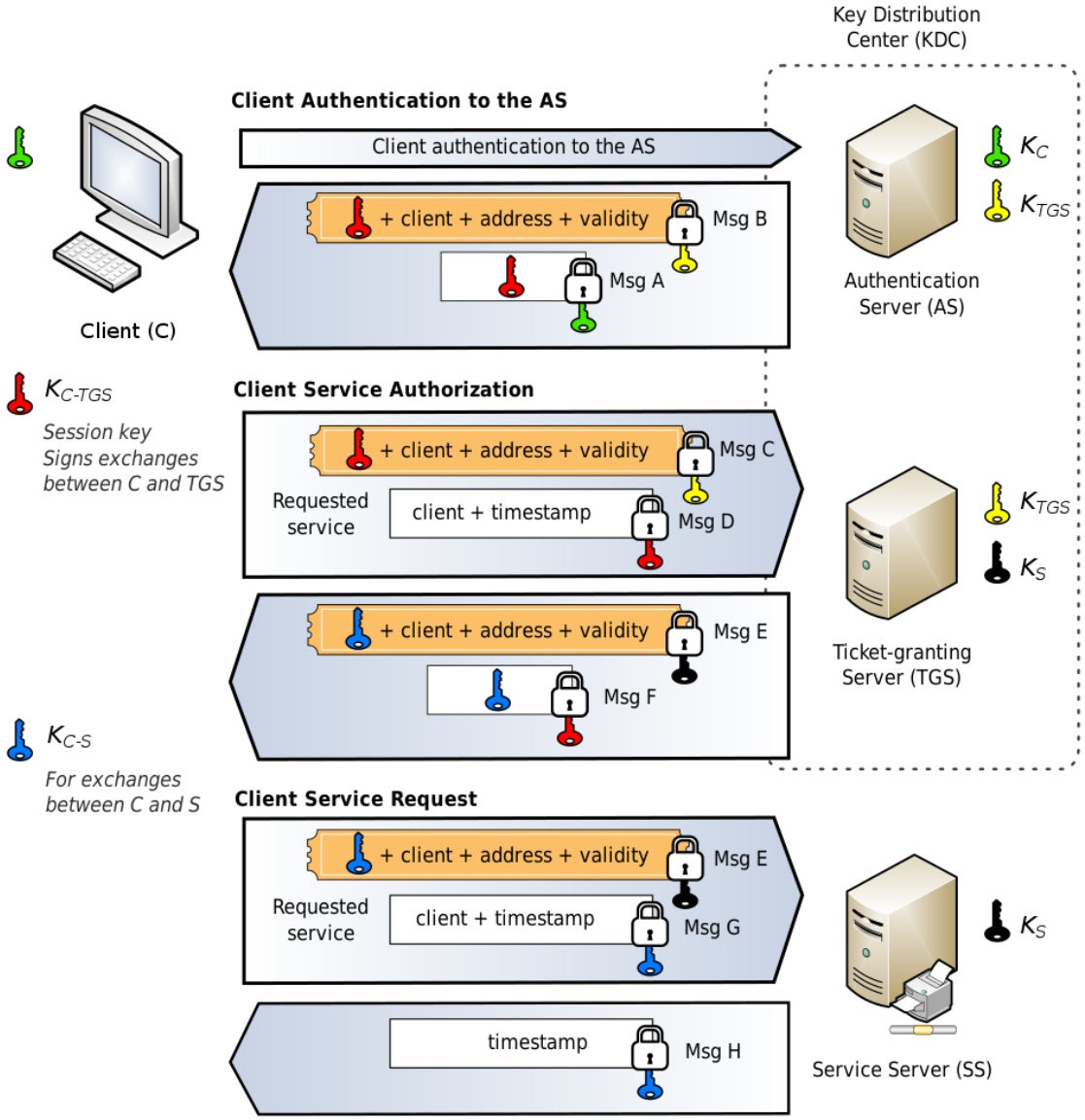
אני מדברת כמובן על **Kerberos Delegation**.

ראשית, חשוב לי לציין שכלל המידע במאמר לקוח ממספר מאמרים מעוררי השראה, שכמובן אצרף אליהם קישור בסוף. מטרת המאמר היא לעזור "להיכנס לענייינים" בנושא כל כך מסובך ומלא במידע, כך שלאחר הקריאה יוכל הקורא לפנות למאמרים המקוריים ולהבין אותם ביתר קלות.

כמו כן, אשתמש הרבה במילה "דלגציה", כי לעבור כל פעם לאנגלית ולכתוב Delegation זה לא ריאלי, ומבדיקה ב-Google Translate, החלופה העברית היא אפילו מוצלחת פחות: "מִשלַחַת", אז... סליחה מראש ☺.

Recap זריז

נבצע מעבר מהיר (מאוד מהיר, מיועד למי שמכיר את הפרוטוקול טוב) על תהליך האימות באמצעות Kerberos. אלו השלבים:



[לקוח מערך הויקיפדיה: Kerberos]

שלב א'

הלקוח ישלח ל-KDC (ובתוכו, ל-Authentication Server) בקשה ל-TGT, הבקשה תכיל את שם המשתמש (לא מוצפן) + השירות המבוקש (SPN), במקרה של בקשת TGT השירות המבוקש הוא krbtgt.

```
Kerberos
    as-req
        pvno: 5
        msg-type: krb-as-req (10)
        padata: 1 item
            PA-DATA PA-REQ-ENC-PA-REP
                padata-type: kRB5-PADATA-REQ-ENC-PA-REP (149)
                    padata-value: <MISSING>
        req-body
            Padding: 0
            kdc-options: 40000010 (forwardable, renewable-ok)
            cname
                name-type: kRB5-NT-PRINCIPAL (1)
                cname-string: 1 item
                    CNameString: Administrator
            realm: SAMBA.EXAMPLE.COM
            sname
                name-type: kRB5-NT-SRV-INST (2)
                sname-string: 2 items
                    SNameString: krbtgt
                    SNameString: SAMBA.EXAMPLE.COM
            till: 2015-01-30 15:17:09 (UTC)
            nonce: 1579110570
            etype: 3 items
                ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
                ENCTYPE: eTYPE-AES128-CTS-HMAC-SHA1-96 (17)
                ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)
```

תחילה, הלקוח יקבל חזרה מה-KDC שגיאה: KDC_ERR_PREAUTH_REQUIRED, זאת משום שהלקוח לא שלח Authenticator (ולידציה שהלקוח הוא אכן הלקוח ולא מדובר בתרחיש MITM כלשהו). ה-Authenticator הוא פשוט timestamp מוצפן עם הסיסמה של הלקוח.

```
Kerberos
    krb-error
        pvno: 5
        msg-type: krb-error (30)
        ctime: 1981-02-06 08:30:31 (UTC)
        stime: 2015-01-29 15:17:03 (UTC)
        susec: 634560
        error-code: eRR-PREAUTH-REQUIRED (25)
        crealm: SAMBA.EXAMPLE.COM
        cname
            name-type: kRB5-NT-PRINCIPAL (1)
            cname-string: 1 item
                CNameString: LOCALDC$
        realm: SAMBA.EXAMPLE.COM
        sname
            name-type: kRB5-NT-SRV-INST (2)
            sname-string: 2 items
                SNameString: krbtgt
                SNameString: SAMBA.EXAMPLE.COM
        e-text: NEEDED_PREAUTH
```

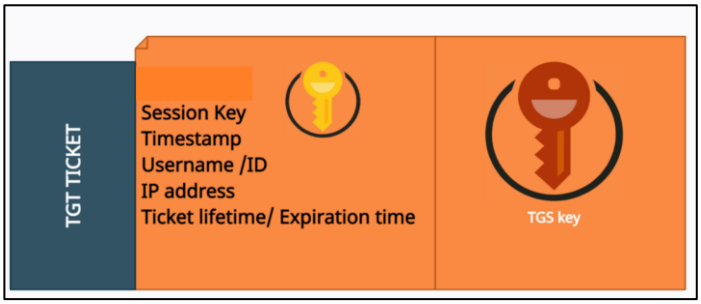
לאחר מכן הלקוח ישלח AS_REQ שנית, הפעם בצירוף ה-Authenticator:

```
Kerberos
  as-req
      pvno: 5
      msg-type: krb-as-req (10)
    padata: 3 items
      PA-DATA PA-FX-COOKIE
        padata-type: kRB5-PADATA-FX-COOKIE (133)
            padata-value: 4d4954
      PA-DATA PA-ENC-TIMESTAMP
        padata-type: kRB5-PADATA-ENC-TIMESTAMP (2)
          padata-value: 3041a003020112a23a0438ada57bc9613417ec74ffa3c5c3...
      PA-DATA PA-REQ-ENC-PA-REP
        padata-type: kRB5-PADATA-REQ-ENC-PA-REP (149)
            padata-value: <MISSING>
    req-body
        Padding: 0
      kdc-options: 00010010 (canonicalize, renewable-ok)
      cname
          name-type: kRB5-NT-PRINCIPAL (1)
        cname-string: 1 item
            CNameString: LOCALDC$
      realm: SAMBA.EXAMPLE.COM
      sname
          name-type: kRB5-NT-SRV-INST (2)
        sname-string: 2 items
            SNameString: krbtgt
```

שלב ב'

במידה וה-KDC אכן מזהה את הלקוח (מצליח לפענח את ה-Authenticator), הוא יחזיר לו בתשובה, AS_REP, אשר יכיל את ה-Session Key שבאמצעותו יכול הלקוח לתקשר מול ה-TGS (מוצפן בסיסמא של הלקוח), כמו כן, הוא יחזיר לו TGT - כרטיס המוצפן עם הסיסמא של ה-TGS.

באמצעות ה-TGT הלקוח בעצם מוכיח שהזדהה בהצלחה מול ה-KDC:



שלב ג'

הלקוח יפענח את ה-Session Key באמצעות הסיסמא שלו וישלח בקשה ל-TGS, הבקשה תכיל:

1. Authenticator בדיוק כמו הבקשה ל-TGT
2. את הכרטיס המוצפן בסיסמא של ה-TGS שהתקבל בשלב א' (TGT) והשירות המבוקש (SPN)

```
▼ Kerberos
  ▼ tgs-req
      pvno: 5
      msg-type: krb-tgs-req (12)
    ▼ padata: 2 items
      ▼ PA-DATA PA-TGS-REQ
        ▼ padata-type: kRB5-PADATA-TGS-REQ (1)
          ▶ padata-value: 6e8204b0308204aca003020105a10302010ea20703050000...
      ▼ PA-DATA PA-FX-FAST
        ▼ padata-type: kRB5-PADATA-FX-FAST (136)
          ▶ padata-value: a081d13081cea1173015a003020110a10e040c57b7dc7e96...
    ▼ req-body
        Padding: 0
      ▶ kdc-options: 40810000 (forwardable, renewable, canonicalize)
        realm: SAMBA.EXAMPLE.COM
      ▼ sname
          name-type: kRB5-NT-PRINCIPAL (1)
        ▼ sname-string: 2 items
            SNameString: ldap
            SNameString: localdc.samba.example.com
        till: 2015-01-30 01:17:09 (UTC)
        nonce: 1422544629
      ▼ etype: 3 items
          ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
          ENCTYPE: eTYPE-AES128-CTS-HMAC-SHA1-96 (17)
          ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)
```

שלב ד'

ה-TGS יפענח את ה-TGT עם הסיסמא שלו, יוודא את ה-Authenticator ולאחר מכן יחזיר ללקוח:

1. Service Ticket שיהיה מוצפן בסיסמא של השירות המבוקש
2. חבילה נוספת, המוצפנת ב-Session Key שהושג קודם לכן, בתוכה יהיה שם השירות, תוקף הכרטיס ו-Session Key

שלב ה'

הלקוח מפענח את החבילה, ושולח 2 בקשות ל-Service:

1. Authenticator רק במקום להיות מוצפן בסיסמא של המשתמש הוא יוצפן ב-Session Key
2. ה-TGS שמוצפן בסיסמא של ה-Service

שלב ו'

ה-Service יפענח את ה-TGS, יחלץ ממנו את ה-Session Key ויפענח את ההצפנה של ה-Authenticator. במידה והכל תקין הוא שולח חזרה למשתמש Authenticator (שיכיל את שם האפליקציה ו-Timestamp) מוצפן עם ה-Session Key.

הלקוח מאמת את ה-Authenticator של השרת וההזדהות הושלמה בהצלחה.

נקודות חשובות:

- תיאור ההזדהות הזה הוא חלקי ביותר, איני מפרטת על כלל השדות בכרטיסים השונים שחלקם חשובים ביותר כמו PAC, תכונות הכרטיס (Forwardable, Renewable וכו'). בחלקם אגע יותר בהמשך המאמר.

- כאשר תיארתי מידע שמגיע בכרטיסים, לא מדובר בכלל המידע אלא במידע שמצאתי יותר רלוונטי למאמר. למשל, ה-Authenticator מכיל בתוכו שדות נוספים כמו Client ID, Checksum ועוד.

Delegation

תחילה נסביר מה היא דלגציה באופן כללי, לצורך הסברת המונח אשתמש בדוגמא:

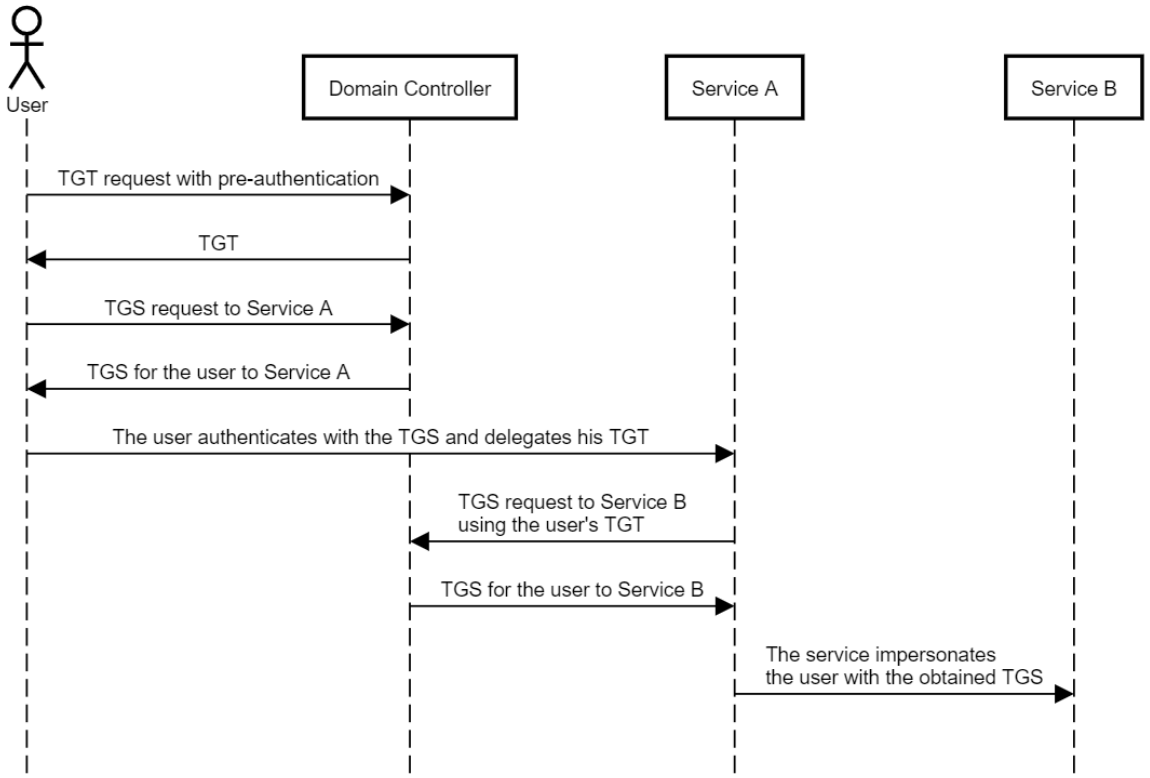
1. אליס מתחברת למשתמש שלה בבנק דרך האתר.
2. כעת, היא רוצה לבדוק כמה כסף יש לה בחשבון, כמובן שהמידע הזה נמצא במאגר המידע ברשת הפנימית של הבנק, ורק המשתמש של אליס רשאי לקרוא כמה כסף יש לה בחשבון.
3. אליס היא אישה עסוקה והיא לא מתחברת לרשתות פנימיות של בנקים בזמנה הפנוי ועושה שאילתות כדי לדעת כמה כסף יש לה בחשבון. לכן, תפקיד האתר הוא לגשת לשרת בשבילה, לבצע את הפעולות הללו ולהציג לה מה סכום הכסף בחשבון.
4. אך על מנת שהאתר יוכל למשוך את המידע מהמאגר הוא צריך את ההרשאות של אליס. לכן, אליס תאשר לאתר לגשת בשמה למאגר המידע ולבדוק מה הסכום בחשבון.

לסיכום, בתהליך הדלגציה, השרת מבצע פעולות מול שירותים שונים (SPN-ים) בשם חשבון אחר, על מנת לבצע פעולות שרק מבקש השירות רשאי לבצע.

ישנן מספר שיטות לבצע האצלת סמכויות שכזו, נעבור על כל אחת ונציג את הבעיות העולות ממנה.

Unconstrained Delegation

בתהליך זה, אליס מזדהה ל-Service A באמצעות Kerberos, בנוסף, היא מעבירה את ה-TGT שלה אליו, והוא מבצע בשמה, באמצעות ה-TGT, בקשת TGS ל-Service B:



- על ה-UAC של השירות שהולך להשתמש ב-Unconstrained Delegation, במקרה שלנו Service A, להיות בעל ההגדרה: TRUSTED_TO_AUTH_FOR_DELEGATION:

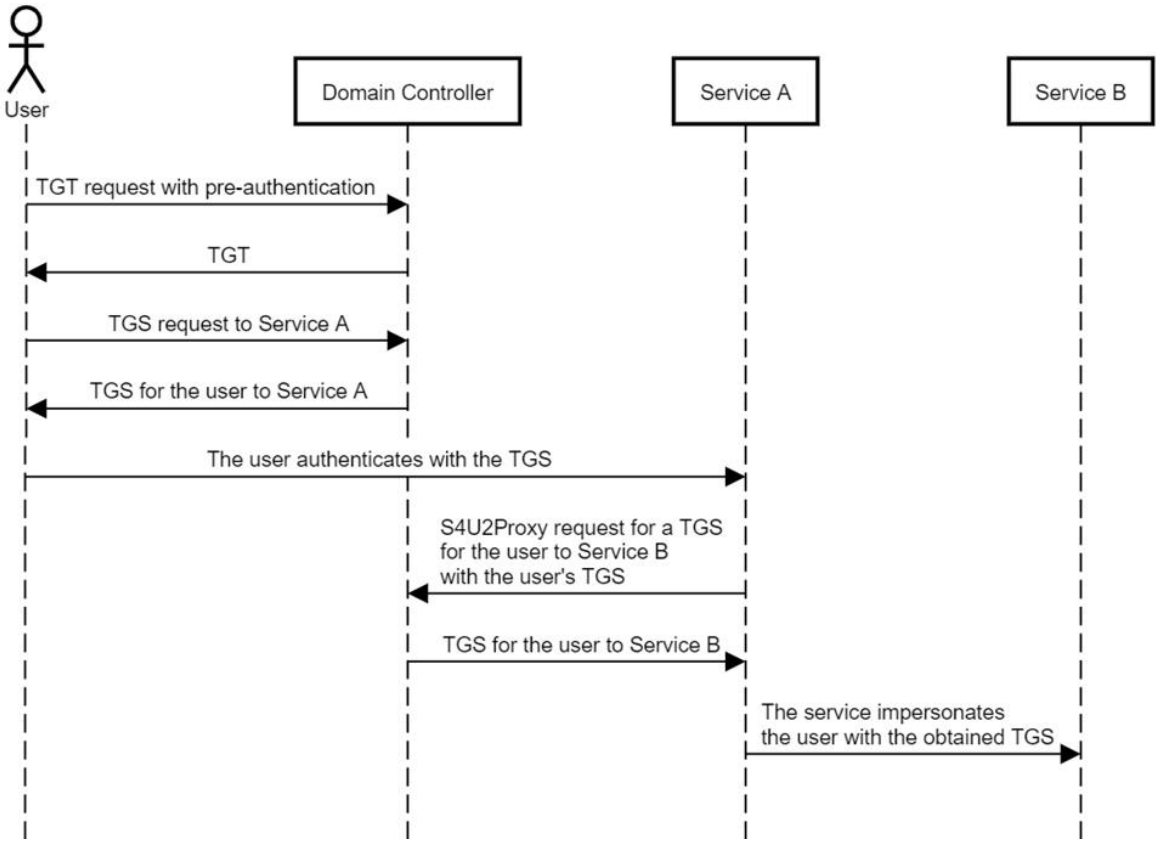
```
PS C:\Users\Administrator> Get-DomainComputer -Unconstrained -Properties useraccountcontrol,dnshostname | fl

dnshostname        : DC01.dev.biz.local
useraccountcontrol : SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION

dnshostname        : WINWS-01.dev.biz.local
useraccountcontrol : WORKSTATION_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION
```

- נוכל כמובן לזהות כאן את הפערים האבטחתיים, כמו למשל, אם יש ברשותי את ה-TGT של אליס, מה מונע ממני להתחזות לה בפעולות בצורה לא לגיטימית? נדון בכך בהמשך.

Constrained Delegation - s4u2proxy

בתהליך זה, נמנע מהמשתמש להעביר את ה-TGT שלו ל-Service A, במקום זה, Service A ישתמש בפיצ'ר שנקרא s4u2proxy המאפשר לו לבקש TGS בשם אליס ל-Service B מבלי להחזיק ב-TGT שלה.

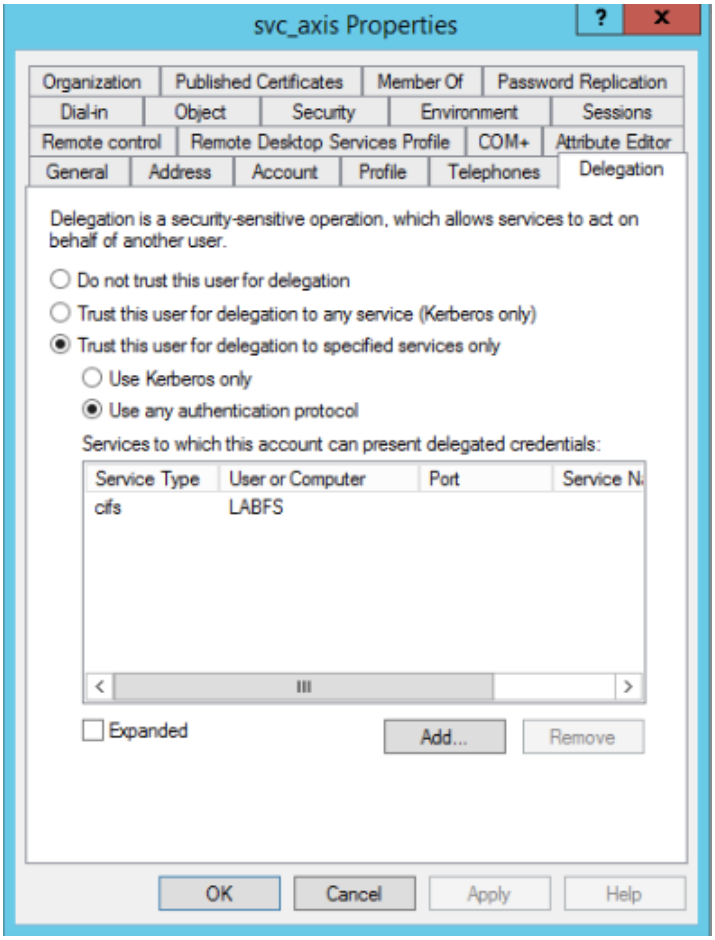
Service A מבצע בקשת s4u2proxy, הוא בעצם מעביר ל-DC כרטיס TGS מהמשתמש עבורו, ומבקש כרטיס TGS עבור המשתמש ל-Service B.



- service A רשאי להשתמש ב-s4u2proxy ויקבל כרטיס אך ורק אם הוא מוגדר מראש ב-service שיכול לבצע Constrained Delegation.

על מנת לדעת אם Service A אכן בעל הרשאות אלה, נוכל לבדוק ב-AD את הערך:

msDS-AllowedToDelegateTo



או כמובן עם Powershell. נוכל לראות בערך בדיוק לאילו SPN-ים השירות יכול לבצע דלגציה. על ה-UAC להכיל את הערך TRUSTED_TO_AUTH_FOR_DELEGATION:

```
PS C:\Users\Administrator> Get-DomainComputer -Unconstrained -Properties useraccountcontrol,dnshostname | fl

dnshostname         : DC01.dev.biz.local
useraccountcontrol  : SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION

dnshostname         : WINWS-01.dev.biz.local
useraccountcontrol  : WORKSTATION_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION
```

על מנת לערוך הרשאות אלו יש צורך במשתמש domain admin בעל הרשאת- SeEnableDelegation.

כדי להשתמש ב-s4u2proxy, ה-TGS המקורי של המשתמש כלפי ה-service חייב להכיל את הדגל-forwardable, עם זאת, נראה בהמשך כי הטענה הזו לא נכונה במלואה.

הסבר על הדגל מ-MS-SFU:

forwardable: A flag, as specified in [RFC4120] section 2.6, used in an **S4U2self** KRB_TGS_REQ message to request that the resulting **service ticket** be marked as forwardable, allowing it to be used in a subsequent **S4U2proxy** KRB_TGS_REQ message.

s4u2self

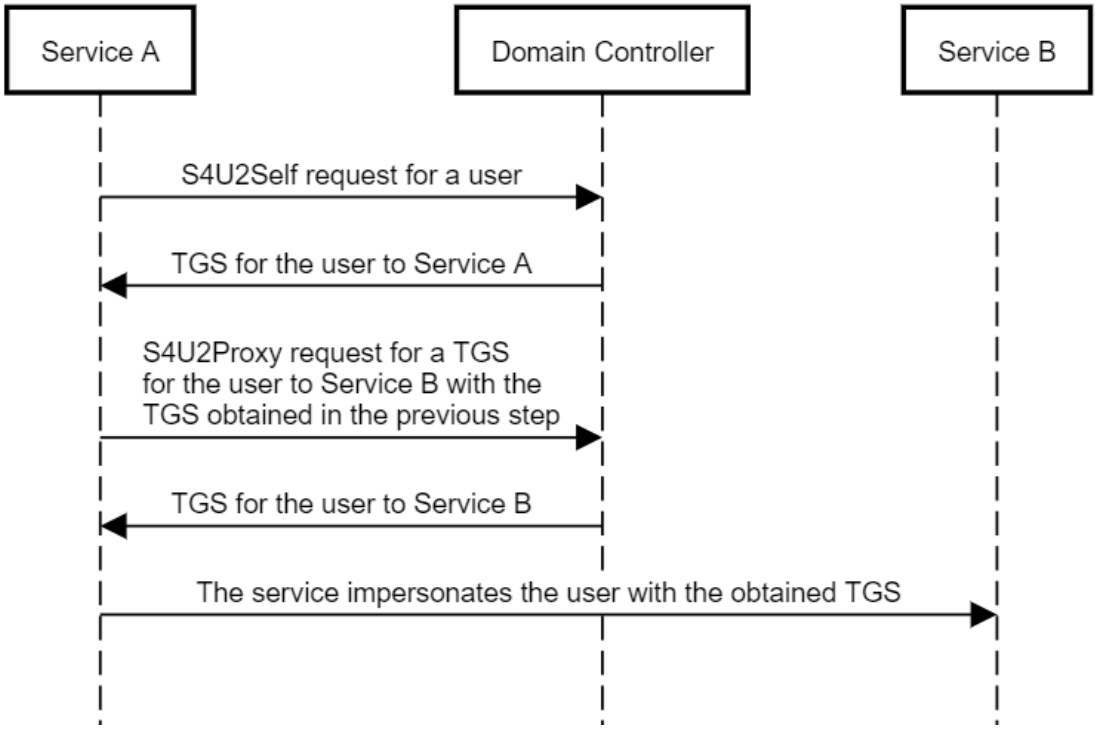
לא מדובר בסוג נוסף של דלגציה, אלא עוד פיצ'ר של Kerberos על מנת להקל על החיים (לא ברור אם מדובר בחיים של התוקף או של המשתמשים).

הרעיון:

כדי ששירות יוכל להשתמש ב-s4u2proxy הוא צריך להציג TGS לעצמו מהמשתמש אליו הוא מתכוון להתחזות. זאת אומרת, סוג של הוכחה שהמשתמש אכן הזדהה אליו לפני שהוא מתחזה לו. אבל, מה אם בתרחיש שלנו המשתמש המדובר לא הזדהה לשרת באמצעות קרברוס? למשל, מה אם הוא הזדהה אליו דרך שם משתמש וסיסמא לממשק WEBי שאינם דומיינים? או אפילו באמצעות NTLM.

במקרה כזה, לשירות אין את ה-TGS מהמשתמש והוא אינו יכול לבצע s4u2proxy.

לכן, על מנת שמקרים כאלו כן יוכלו להתקיים, נוצר הפיצ'ר s4u2self, באמצעותו, יוכל השירות קודם לבקש TGS בשם המשתמש לעצמו, ולאחר מכן להשתמש ב-TGS הזה על מנת לבצע s4u2proxy.



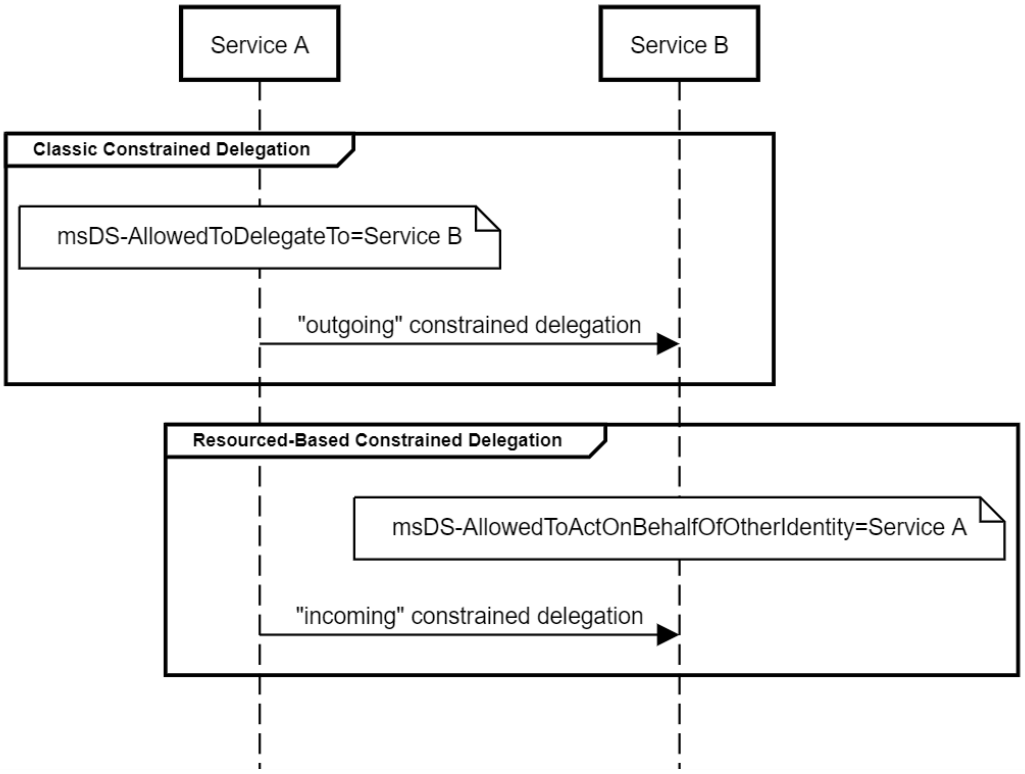
- כפי שניתן להבין, הפיצ'ר הזה מאפשר התחזות למשתמש ממש מאפס(או כפי שאלעד שמיר אמר ומאוד אהבתי - out of thin air). לכן, שירות יוכל להשתמש בו אך ורק אם הדגל-TrustedToAuthForDelegation מאופשר ל-service account שמשתמש בו.

- אם נשתמש ב-s4u2self אך ה-service שלנו לא מוגדר כ-TrustedToAuthForDelegation, נקבל בכל זאת TGS אך הוא לא יהיה forwardable ולכן לא נוכל להשתמש בו ב-s4u2proxy(זו היא בעיית ה-double hop המוכרת).

Resource-Based Constrained Delegation(RBCD)

כפי שראינו, על מנת להחליט מי יכול לבצע דלגציה למי יש צורך בהרשאות מאוד גבוהות, על מנת להעניק קצת יותר "חופש", נוצר Resource-Based Constrained Delegation, המאפשר ל-service-ים להחליט אילו חשבונות יכולים לבצע דלגציה אליהם.

השיטה הזו דומה מאוד ל-Constrained Delegation, ההבדל הוא שהכיוון הוא הפוך. זאת אומרת, במקום שיוגדר לשירות כלפי מי הוא יכול לבצע דלגציה (באמצעות msDS-AllowedToDelegateTo), השירות מגדיר לעצמו מי יוכל לבצע דלגציה אליו (באמצעות msDS-AllowedToActOnBehalfOfOtherIdentity):



- לא דורש הרשאות גבוהות, רק הרשאות GenericWrite
- נקודה מעניינת שאלעד שמיר גילה, נוכל להשתמש בכרטיס TGS שאינו forwardable (שנוכל להשיג באמצעות s4u2proxy כפי שציינתי למעלה) ולבצע resource-based constrained delegation והדלגציה תעבוד בכל מקרה! מדובר רק בדלגציה הזו, לשאר הטכניקות זה לא יעבוד.

מתקפות מבוססות דלגציה

כעת אפרט על מספר מתקפות המבוססות על סוגי הדלגציה השונים שראינו. אציין כי יש עוד הרבה מאוד מתקפות ואלו הן דוגמאות בסיס שיעזרו להבין את המתקפות המסובכות יותר במידה ותרצו בכך.

Abusing Unconstrained Delegation

נוכל להשתמש ב-Unconstrained Delegation לצורך גניבת TGT.

התרחיש: "נשכנע" שירות להתאמת מולינו ב-Kerberos ולהעביר לנו את ה-TGT לצורך דלגציה. דוגמא לכך היא שימוש ב-Printer Bug.

הסבר: נשתלט על מחשב בעל הרשאות Unconstrained Delegation (דיפולטית תמיד ב-DC-ים קיימת הרשאה זו ולכן אוהבים להשתמש בהם כקורבן למתקפה זו, וגם כי זה DC ומי לא רוצה להתחזות ל-DC ☺).

פקודת PS פשוטה למציאת חשבונות ברשת בעלי הרשאת דלגציה מתאימה:

```
Get-ADComputer -Filter {TrustedForDelegation -eq $True}
```

נגרום לשירות להזדהות מול המחשב שלנו באמצעות Printer Bug: בשתי מילים, ניצול של השירות spooler על שרת ושימוש בפקודת RPC (בדרך כלל RpcRemoteFindFirstPrinterChangeNotificationEx) כדי לגרום לשרת להתאמת מולינו. כמובן שלצורך התרחיש הספציפי הזה צריך לפעול על הקורבן שירות מדפסות.

כך בעצם נגנוב את ה-TGT של הקורבן ונוכל להשתמש בו למתקפות שונות. כמובן שנוכל לגרום לשירות להזדהות אלינו בהמון דרכים נוספות:

- Responder
- ARP Poisoning
- Petit Potam (שעובד בצורה דומה רק במקום פונקציות RPC של מדפסות הוא משתמש בפונקציות RPC של קבצים) וחברים נוספים.
- הכלי Rubeus מבצע את כל גניבת הכרטיסים (כתוב ב-#C)
- נוכל להשתמש בכלי printerbug.py כדי להשמיש את החולשה ולגרום לקורבן להזדהות אלינו ב-RPC.

Privilege Escalation Using s4u

כאשר שירות ישתמש ב-s4u2self הוא יעביר ל-DC:

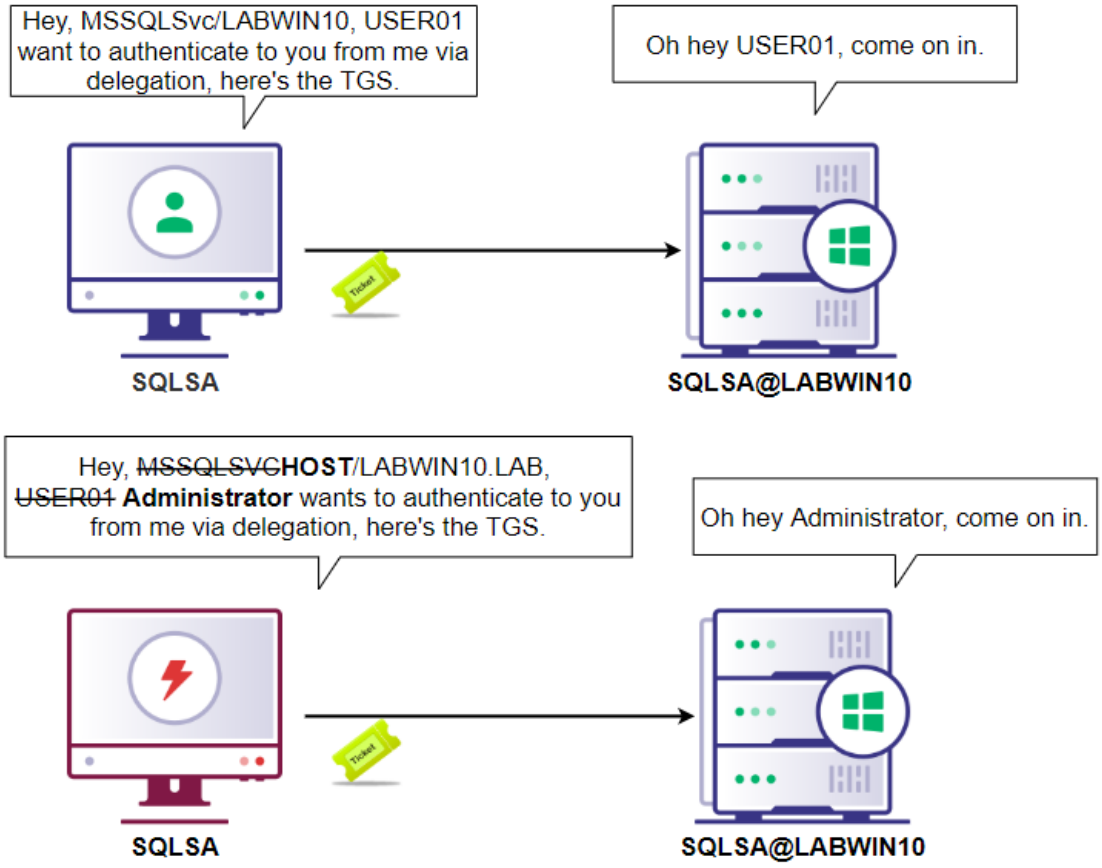
1. כרטיס TGT של השירות (עצמו)
2. בקשה ל-TGS עם משתמש אחר (למשל Administrator)

לאחר שיש בידינו TGT, נשלח בקשת s4u2self למשתמש חזק בצירוף ה-TGT, וזהו, קיבלנו TGS לעצמינו עם משתמש חזק! מכאן נוכל להשתמש בו בשביל s4u2proxy או פשוט להסלים הרשאות על המחשב שלנו.

- נקודה מעניינת: נניח והצלחנו להשיג TGS לשירות SQL באמצעות s4u2proxy אבל אנו רוצים לגשת לקורבן בשירות אחר (אולי HTTP כדי להשתמש ב-WinRM). נוכל לערוך את הכרטיס שכן ה-SPN אינו מוצפן ואנו על אותו מחשב כך שהסיסמא של השירותים השונים אינה משתנה והכרטיס ישאר תקף.

- HOST SPN הוא בעצם SPN המכיל בתוכו הרבה סוגי שירותים שונים, נוכל לראות את השירותים באמצעות פקודת PS, לכן הרבה פעמים נשנה את ה-SPN ל-Host במתקפה זו.

```
PS C:\Users\Administrator> Get-ADObject -Identity "CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC=AD
SEC,DC=LOCAL" -properties sPNMappings


DistinguishedName : CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC=ADSEC,DC=LOCAL
Name              : Directory Service
ObjectClass       : nTDSService
ObjectGUID        : 48747387-b6D7-4eD2-8672-10792e79864b
sPNMappings       : {host=alerter,appmgmt,cisvc,clipsrv,browser,dhcp,dnscache,replicator,eventlog,eventsystem,policyage
                    nt,oakley,dmserver,dns,mcsvc,fax,msiserver,ias,messenger,netlogon,netman,netdde,netddedsm,nmagent,p
                    lugplay,protectedstorage,rasman,rpclocator,rpc,rpcss,remoteaccess,rsvp,samss,scardsvr,scesrv,seclog
                    on,scm,dcom,cifs,spooler,snmp,schedule,tapisrv,trksvr,trkwks,ups,time,wins,www,http,w3svc,iisadmin,
                    msdtc}
```



אציג תרחיש פשטני מאוד ולאחר מכן הרחבה מסוימת שלו. נחפש באמצעות PS חשבון שיש שירותים שיכולים להזדהות אליו ב-RBCD:

```
PS C:\tmp> Get-DomainComputer | Where-Object {$_."msDS-AllowedToActOnBehalfOfOtherIdentity" -ne $null}

logoncount                              : 31
badpasswordtime                         : 12/31/1600 7:00:00 PM
distinguishedname                       : CN=labaxis,OU=Web-Servers,OU=Servers,OU=Computers,OU=lab,DC=lab,DC=local
objectclass                             : {top, person, organizationalPerson, user...}
badpwdcount                             : 0
lastlogontimestamp                      : 6/19/2019 6:55:09 PM
userprincipalname                       : labaxis@lab.local
objectsid                               : S-1-5-21-1046923091-56574964-3868113300-1130
samaccountname                          : labaxis$
localpolicyflags                        : 0
codepage                                : 0
samaccounttype                          : MACHINE_ACCOUNT
countrycode                             : 0
cn                                      : labaxis
accountexpires                          : NEVER
whenchanged                             : 6/20/2019 9:00:16 PM
instancetype                            : 4
usncreated                              : 17933
objectguid                              : d5bfbce2-363e-441d-b580-cd32acfdf09d
operatingsystem                         : Windows Server 2012 R2 Standard
operatingsystemversion                  : 6.3 (9600)
lastlogoff                              : 12/31/1600 7:00:00 PM
msds-allowedtoactonbehalfofotheridentity : {1, 0, 4, 128...}
objectcategory                          : CN=Computer,CN=Schema,CN=Configuration,DC=lab,DC=local
dscorepropagationdata                   : {6/20/2019 9:00:16 PM, 6/20/2019 8:59:18 PM, 6/20/2019 8:57:20 PM,
                                          6/20/2019 8:44:33 PM...}
serviceprincipalname                    : {TERMSRV/LABAXIS, TERMSRV/labaxis.lab.local, WSMAN/labaxis,
                                          WSMAN/labaxis.lab.local...}
lastlogon                               : 6/21/2019 11:20:52 AM
iscriticalsystemobject                  : False
usnchanged                              : 22503
useraccountcontrol                      : WORKSTATION_TRUST_ACCOUNT, DONT_EXPIRE_PASSWORD
whencreated                             : 6/10/2019 8:50:06 PM
primarygroupid                          : 515
pwdlastset                              : 6/19/2019 9:42:10 PM
msds-supportedencryptiontypes           : 28
name                                    : labaxis
dnshostname                             : labaxis.lab.local
```

נוכל לראות שהערך לא בדיוק ניתן לקריאה, נוכל לקרוא עם PS את הערך הבא (שעל פי האתר של Microsoft הוא התרגום ל-ACL שב-msDS-AllowedToActOnBehalfOfOtherIdentity):

"The PrincipalsAllowedToDelegateToAccount parameter sets the Active Directory object attribute msDS-AllowedToActOnBehalfOfOtherIdentity, which contains an access control list (ACL) that specifies which accounts have permission to delegate credentials to the associated account"

```
PS C:\Users\Administrator.lab> Get-ADComputer labaxis -Properties PrincipalsAllowedToDelegateToAccount

DistinguishedName                   :CN=labaxis,OU=Web-Servers,OU=Servers,OU=Computers,OU=lab,DC=lab,DC=local
DNSHostName                         :labaxis.lab.local
Enabled                             :True
Name                                :labaxis
ObjectClass                         :computer
ObjectGUID                          :d5bfbce2-363e-441d-b580-cd32acfdf09d
PrincipalsAllowedToDelegateToAccount :{CN=svc_tomcat,OU=Service Accounts,OU=IT,OU=Users,OU=lab,DC=lab,DC=local}
SamAccountName                      :labaxis$
SID                                 :S-1-5-21-1046923091-56574964-3868113300-1130
UserPrincipalName                   :labaxis@lab.local
```

נוכל לראות ש-svc_tomcat מורשה להזדהות ב-RBCD לכל service ב-labaxis. לכן במידה ונשלוט במשתמש (יהיה לנו TGS שלו) נשלוט גם ב-labaxis.

הרחבה - שימוש ב-GenericWrtie

נאתר (במקרה שלנו באמצעות הכלי powerview שכתוב ב-PS) משתמש בעל הרשאות Generic Write:

```
PS C:\tmp> $tomcat_sid = "lab\svc_tomcat" | Convert-NameToSid
PS C:\tmp> Get-DomainObjectAcl -SearchBase "LDAP://OU=Computers,OU=lab,DC=lab,DC=local" | Where-Object {$_.SecurityIdent
ifier -match $tomcat_sid}


ObjectDN              : CN=labaxis,OU=Web-Servers,OU=Servers,OU=Computers,OU=lab,DC=lab,DC=local
ObjectSID             : S-1-5-21-1046923091-56574964-3868113300-1130
ActiveDirectoryRights : ListChildren, ReadProperty, GenericWrite
BinaryLength          : 36
AceQualifier          : AccessAllowed
IsCallback            : False
OpaqueLength          : 0
AccessMask            : 131132
SecurityIdentifier    : S-1-5-21-1046923091-56574964-3868113300-1117
AceType               : AccessAllowed
AceFlags              : None
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None
AuditFlags            : None
```

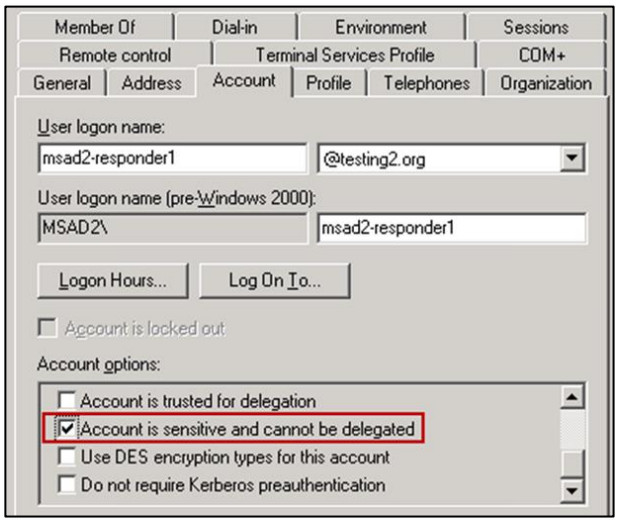
החיפוש כאן מתבצע באמצעות SID והוא משתמש בכלי PowerSploit(הוספתי קישור ל-GIT של הפרויקט עם העמוד הרלוונטי לפונקציה זו), נוכל לראות שחיפשנו את ה-SID של svc_tomcat ולפי התוצאה, הוא בעל הרשאות GenericWrite ל-labaxis.

כעת, נניח ואנו שולטים במשתמש svc_tomcat ובעמדה שנקרא לה host, נוכל להשתמש בהרשאות של svc_tomcat, ולהוסיף ל-labaxis בערך msDS-AllowedToActOnBehalfOfOtherIdentity את host.

לבסוף, נוכל להשתמש ב-s4u2proxy על host ולהעביר את הכרטיס ל-labaxis, וכך להתחזות לאיזה משתמש שנרצה ולבקש איזה שירות שנרצה!

עבודת המגן

במאמר זה אתייחס פחות לפן המניעה, אלא יותר בפן הזיהוי. אם כי חשוב לציין כי קיים קונספט בשם Sensitive User המגדיר כי המשתמש רגיש ולא ניתן לבצע איתו דלגציה מאף סוג (נכון להיום):



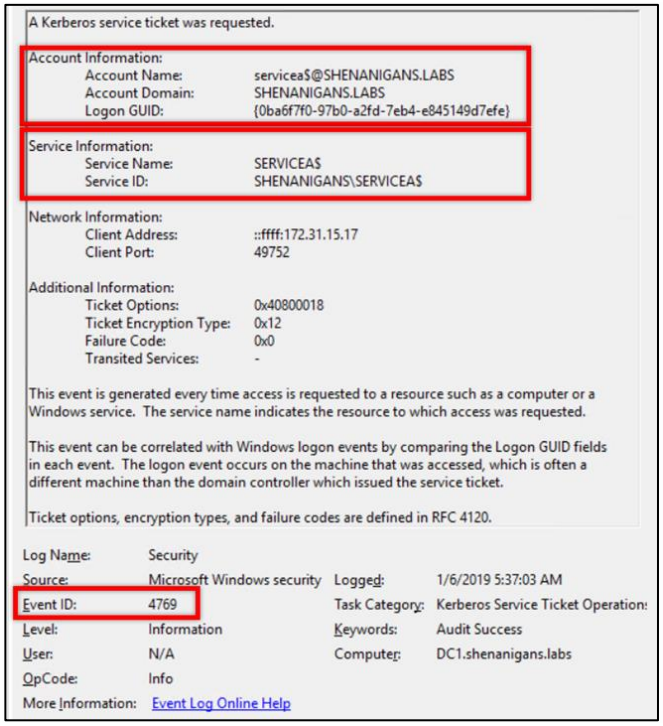
בנוסף אוסיף, שבתור מגן המכיר את הרשת,הייתי ממליצה לבצע מיפוי של האובייקטים בהם מתאפשרת דלגציה מהסוגים השונים, ולבצע מעקב אחר שינויים, מתוך הנחה שלא אמורים להיות הרבה כאלה.

זיהוי s4u2self

נוכל לזהות שימוש ב-s4u2self באמצעות EventID 4769 security log:

"A Kerberos service ticket was requested"

כאשר ה-service information וה-account information מצביעים על אותו חשבון:



זיהוי s4u2proxy

נוכל לזהות שימוש ב-s4u2proxy באמצעות EventID 4769 ב-Security Log:

A Kerberos service ticket was requested

כאשר ה-transited service שנמצא תחת additional information אינו ריק:

A Kerberos service ticket was requested.

Account Information:
Account Name: servicea$@SHENANIGANS.LABS
Account Domain: SHENANIGANS.LABS
Logon GUID: {0ba6f7f0-97b0-a2fd-7eb4-e845149d7efe}

Service Information:
Service Name: SERVICEB$
Service ID: SHENANIGANS\SERVICEB$

Network Information:
Client Address: ::ffff:172.31.15.17
Client Port: 49753

Additional Information:
Ticket Options: 0x40820010
Ticket Encryption Type: 0x12
Failure Code: 0x0
Transited Services: servicea$@SHENANIGANS.LABS

This event is generated every time access is requested to a resource such as a computer or a Windows service. The service name indicates the resource to which access was requested.

This event can be correlated with Windows logon events by comparing the Logon GUID fields in each event. The logon event occurs on the machine that was accessed, which is often a different machine than the domain controller which issued the service ticket.

Ticket options, encryption types, and failure codes are defined in RFC 4120.

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	1/6/2019 5:37:03 AM
Event ID:	4769	Task Category:	Kerberos Service Ticket Operations
Level:	Information	Keywords:	Audit Success
User:	N/A	Computer:	DC1.shenanigans.labs
OpCode:	Info		
More Information:	Event Log Online Help		

זיהוי שינוי הערך msDS-AllowedToActOnBehalfOfOtherIdentity

נשתמש ב-EventID 5136:

A directory service object was modified

כאשר השדה ldap display name יכיל את הערך msDS-AllowedToActOnBehalfOfOtherIdentity במידה וה-object וה-subject מצביעים על אותו חשבון הדבר עלול להחשיד את הלוג:

A directory service object was modified.

Subject:
Security ID: SHENANIGANS\SERVICEB$
Account Name: SERVICEB$
Account Domain: SHENANIGANS
Logon ID: 0x11AA7F

Directory Service:
Name: shenanigans.labs
Type: Active Directory Domain Services

Object:
DN: CN=SERVICEB,CN=Computers,DC=shenanigans,DC=labs
GUID: CN=SERVICEB,CN=Computers,DC=shenanigans,DC=labs
Class: computer

Attribute:
LDAP Display Name: msDS-AllowedToActOnBehalfOfOtherIdentity
Syntax (OID): 2.5.5.15
Value: Malformed Security Descriptor

Operation:
Type: Value Added
Correlation ID: {cbfa950c-8733-4cfa-ad39-9b8923ebb348}
Application Correlation ID: -

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	1/23/2019 9:44:56 PM
Event ID:	5136	Task Category:	Directory Service Changes
Level:	Information	Keywords:	Audit Success
User:	N/A	Computer:	DC1.shenanigans.labs
OpCode:	Info		
More Information:	Event Log Online Help		

כמו כן, יש באינטרנט אין סוף הצעות לזיהוי מתקפות כמו PrintNightmare (באמצעות לוגים של EventViewer או זיהויים יותר מתקדמים של פקודות RPC), אוסיף קישור בסוף המאמר. בנוסף, תמיד אפשר לחשוב על רעיונות זיהוי נוספים מעניינים, למשל, האם נוכל לזהות מתקפה של שינוי SPN באמצעות מציאת אירוע 4769 ל-SPN מסוים אך גישה לאותו HOST בשירות אחר? דוגמה: 4769 עם SPN של CIFS אבל תקשורת לעמדה בפרוטוקול HTTP.

סיכום

אני מקווה שתרמתי להבנה כללית של הקונספט המעניין שנקרא Delegation בסביבת Domain. כפי שכתבתי, המתקפות והגילויים בתחום זה מתפתחים מיום ליום, ויש עוד הרבה איפה להעמיק! בנוסף, למי שמעוניין להבין את הנושא לעומק, לצורך תקיפה או לצורך יצירת יכולות זיהוי, אני יותר מממליצה לעבור על מקורות המידע שאוסיף!

מקורות מידע

המאמר העיקרי ששימש לכתיבת המאמר, נכתב על ידי אלעד שמיר. מכיל עוד הרבה הסברים על קונספטים ומתקפות, ממליצה בחום לקרוא:

https://shenaniganslabs.io/2019/01/28/Wagging-the-Dog.html

עמוד ה-git של הכלי Powerview:

https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1

השמשה של כלל המתקפות במאמר ומתקפות נוספות:

https://www.guidepointsecurity.com/blog/delegating-like-a-boss-abusing-kerberos-delegation-in-active-directory/

לוגים רלוונטים לציד של שימוש ב-printer nightmare ו-Unconstrained delegation:

https://posts.specterops.io/hunting-in-active-directory-unconstrained-delegation-forests-trusts-71f2b33688e1

מאמר מקיף ומצוין על מתקפות בפרוטוקול Kerberos:

https://posts.specterops.io/kerberosity-killed-the-domain-an-offensive-kerberos-overview-eb04b1402c61

דרך קישור זה תוכלו להגיע לגרסא האחרונה של MS-SFU:

https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-sfu/3bff5864-8135-400e-bdd9-33b552051d94

דרך קישור זה תוכלו להגיע לגרסא האחרונה של ה-RFC של NTLM (לאמיצים בלבד):

https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-nlmp/b38c36ed-2804-4868-a9ff-8dd3182128e4