Technologic Papers

Golden SAML

מאת שקד ריינר

הקדמה

בזמן בו יותר ויותר ארגונים מעבירים את תשתיותיהם לענן, סביבת ה-AD (Active Directory) של ארגון היא לא עוד הסמכות העליונה לזיהוי ואימות משתמשים. סביבת AD יכולה כעת להיות חלק ממשהו גדול יותר - פדרציה (Federation).

סביבת פדרציה הינה סביבה בה ישויות מחשב (בהן AD למשל) מבססות ביניהן יחסי אמון, על פי תקנים מוסכמים מראש. לדוגמא, משתמש AD כחלק מסביבת פדרציה, יכול ליהנות מיתרונות SSO (Single Sign On) כאשר ייגש לכל הסביבות הנוספות החברות בפדרציה זו. בסביבה מסוג זה, תוקף כבר לא יסתפק רק בשליטה ב-AD, אלא ישאף להגיע לשליטה מלאה בכל המערכות השותפות בפדרציה.

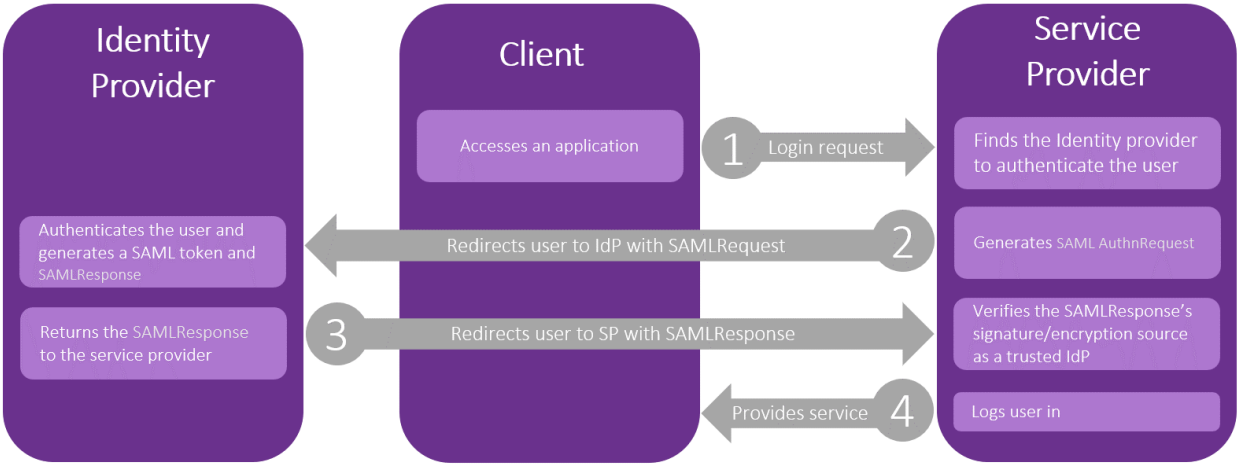
במאמר זה נלמד מהו Golden SAML, המאפשר לתוקף לייצר "אובייקט הזדהות" - SAMLResponse איתו ניתן להתחבר לכל שירות בפדרציה. על ידי ניצול טכניקה זו, תוקף יכול לקבל גישה לשירות התומך בהזדהות SAML, עם זהות והרשאות לפי בקשתו. השם שקיבלה טכניקה זו עשוי להזכיר שם של מתקפה אחרת הנקראת Golden Ticket שהוצגה ע"י בנג'מין דלפי המוכר במיוחד בזכות כתיבתו את mimikatz. דמיון זה לא קיים במקרה, שכן הרעיון מאחורי שתי המתקפות זהה. Golden SAML מאפשר לתוקפים ליהנות מחלק מיתרונותיו של Golden Ticket בסביבת פדרציה. במסגרת פרסום המחקר, שחררתי כלי POC ב-GitHub הנקרא shimit.

נקדים ונאמר ש-Golden SAML אינו חולשה, אלא טכניקת Post Exploitation. טכניקת זו לא מסתמכת על פגיעות ב-SAML 2.0, AWS, AD FS או כל שירות אחר.

Golden SAML מאפשר לתוקף לשמר אחיזה בצורה חשאית בפדרציה. בנוסף, יכול תוקף להרחיב את אחיזתו מסביבת on-premise של ארגון לסביבת הענן שלו בפדרציה (במידה והשירות המזהה ומאמת משתמשים של הארגון קיים on-premise, עוד עליו בהמשך). נתחיל עם הסבר על הצדדים הפעילים, הדרך בה הם מתקשרים והיחסים ביניהם.

SAML

אחד התקנים המשמשים למימוש יחסי האמון בפדרציה הוא SAML. **S**ecurity **A**ssertion **M**arkup **L**anguage הוא סטנדרט פתוח המשמש להחלפת מידע אימות בין ישויות מחשוב. ההודעות המוחלפות ב-SAML מבוססות מסמכי XML. הצדדים הפעילים בהזדהות באמצעות SAML נקראים Identity Provider (IdP) ו-Service Provider (SP). כפי שמרמז שמם, IdP מספק מידע זיהוי ואימות של ישויות בפדרציה, ואילו ה-SP מספק שירותים לישויות אלה. ניתן להקביל זאת לחלוקה דומה שמתקיימת בסביבת AD - Domain Controller מספק מידע הזדהות ואימות של משתמשים, ואילו שרתים אחרים מספקים למשתמשים אלה שירותים (File Servers, Exchange Servers וכו'). התרשים הבא מתאר תהליך התחברות SAML לגיטימי בפדרציה:



1. המשתמש ניגש לשירות מסוים (SP) - דוגמא לשירות יכולה להיות AWS Console, vSphere Web Client וכדומה.
2. ה-SP מזהה לאיזה IdP יש להפנות את המשתמש, מייצר SAML AuthnRequest ומפנה את המשתמש אליו.
3. ה-IdP מזהה ומאמת את המשתמש, מייצר SAMLResponse (אובייקט חתום המכיל את זהות המשתמש) ומפנה את המשתמש חזרה ל-SP יחד עם אובייקט ההזדהות.
4. ה-SP מוודא את אמינות ה-SAMLResponse ע"י וידוא החתימה ומחבר את המשתמש לשירות המבוקש.

על מנת לבצע את המתקפה בהצלחה, על התוקף לבצע בעצמו את שלב 3 המתואר בתרשים. תחילה, נלמד קצת יותר על מבנה ה-SAMLResponse.

SAMLResponse הינו האובייקט אותו מעביר ה-IdP אל ה-SP (באמצעות המשתמש) בתהליך ההזדהות. אובייקט זה מכיל את כל המידע על זהותו של המשתמש - שם המשתמש, קבוצות, הרשאות וכדומה.

המבנה הכללי של SAMLResponse נראה כך:

```
<samlp:Response ID="[id]" Version="2.0" IssueInstant="[timestamp]"
Destination="[SP]" Consent="urn:oasis:names:tc:SAML:2.0:consent:[consent]"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
    <Issuer xmlns="urn:oasis:names:tc:SAML:2.0:assertion">[issuer]</Issuer>
    <samlp:Status>
        <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:[status]" />
    </samlp:Status>
    <Assertion ID="[id]" IssueInstant="[timestamp]" Version="2.0"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
        <Issuer>[IdP]</Issuer>
        <Subject>
            <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">[user]</NameID>
            <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                <SubjectConfirmationData NotOnOrAfter="[confirm_not_on_after]"
Recipient="[recipient]" />
            </SubjectConfirmation>
        </Subject>
        <Conditions NotBefore="[timestamp]" NotOnOrAfter="[timestamp]">
            <AudienceRestriction>[audience]</AudienceRestriction>
        </Conditions>
        <AttributeStatement>[attributes]</AttributeStatement>
        <AuthnStatement AuthnInstant="[timestamp]"
SessionIndex="[session_index]">
            <AuthnContext>
                <AuthnContextClassRef>[IdP]</AuthnContextClassRef>
            </AuthnContext>
        </AuthnStatement>
    </Assertion>
</samlp:Response>
```

ה-Assertion בתוך ה-SAMLResponse יכולה להיות חתומה או מוצפנת על ידי המפתח הפרטי של ה-IdP, כתלות בסוג המימוש. בעזרת החתימה/ההצפנה (והמפתח הציבורי של ה-IdP) מוודא ה-SP שאובייקט ההזדהות אכן נוצר על ידי ה-IdP ביניהם ישנו יחס אמון, וניתן לסמוך על זהות המשתמש המפורטת באובייקט זה.

בדומה ל-Golden Ticket, במידה ותוקף הצליח לשים את ידו על המפתח שחותם את האובייקט שמכיל את זהות המשתמש והרשאותיו (KRBTGT ב-Golden Ticket או token-signing private key ב-Golden SAML), הוא יכול לזייף אובייקטים כאלה (TGT ו-SAMLResponse) ולהתחזות לכל משתמש שקיים בפדרציה.

תוקף יכול לשלוט על כל מאפייני ה-SAMLResponse (שם המשתמש, ההרשאות, תוקף ועוד). בנוסף, ל-Golden SAML היתרונות הבאים:

- ניתן לייצר Golden SAML מכל מקום. התוקף לא צריך להיות חלק מדומיין או פדרציה.
- רלוונטי גם עבור משתמשים בעלי 2 Factor Authentication.
- המפתח הפרטי המשמש את ה-IdP לחתימה אינו מתחלף כברירת מחדל.
- שינוי סיסמא של משתמש לא תשפיע על ה-SAMLResponse.

Golden SAML + AD FS + AWS

בחלק הבא, נציג case study בו תוקף יכול לבצע שימוש ב-Golden SAML על מנת לקבל גישה לא מבוקרת לשירותים הקיימים בפדרציה. את ייצור ה-Golden SAML והשימוש בו אעשה באמצעות כלי שפרסמנו ב-GitHub למטרה זו - shimit.

Active Directory Federation Services או ADFS, הוא שירות Microsoft בסביבת AD המאפשר שיתוף של מידע זיהוי ואימות של משתמשים בין ישויות בפדרציה. שירות זה הוא מימוש של Microsoft ל-IdP, המאפשר למשתמשים ב-Domain להשתמש בזהות שלהם על מנת לגשת לשירותים חיצוניים בסביבת פדרציה.

במידה וישנו חשבון AWS בפדרציה זו, הסומך על הזהויות אותן מקבל משירות ה-ADFS, ולתוקף ישנה גישה לשרת ה-ADFS (זהו תנאי מקדים לטכניקה זו, שכן היא משמשת תוקפים לשמירת האחיזה בארגון והתחמקות מזיהוי, בדומה לתנאי המקדים של גישת Domain Admin במתקפת Golden Ticket), התוקף יכול להשתמש ב-Golden SAML כדי להזדהות בתור כל משתמש ב-AWS, בעל כל הרשאה שיבחר. בשונה מ-Golden Ticket, כדי לממש Golden SAML לתוקף לא צריכה להיות גישה לחשבון Domain Admin או Local Admin בהכרח, אלא רק גישה ל-ADFS Service Account. על מנת להרכיב SAMLResponse בצורה תקינה, על התוקף לדעת את הפרטים הבאים:

- **IdP token-singing private key**
- **IdP public certificate**
- **IdP Name**
- **Role name in AWS**
- **AWS account ID**
- Domain + username
- Role session name in AWS

את הפרטים המודגשים חייב התוקף לדעת על סביבת המטרה, הפרטים האחרים יכולים להיקבע על ידיו באופן שרירותי. איך משיגים את הפרטים האלו? אל המפתח הפרטי של ה-IdP ניתן לגשת מה- ADFS Service Account, הוא מאוחסן תחת ה-Personal Certificate Store שלו (ניתן להשתמש בכלים כמו mimikatz). עבור הפרטים האחרים, ניתן להשתמש בפקודות PowerShell הבאות (להריצן בתור ה- ADFS Service Account):

ADFS Public Certificate:

```
PS > [System.Convert]::ToBase64String((Get-AdfsCertificate | ?
{$_.CertificateType like 'Token-Signing'}).certificate.rawdata)
```

```
PS > (Get-ADFSProperties).Identifier.AbsoluteUri
```

IdP Name:

```
PS > (Get-ADFSRelyingPartyTrust).IssuanceTransformRule # Derived from this
```

Role Name:

לאחר שאספנו את כל הפרטים הדרושים, נצלול ישר לביצוע. תחילה נבדוק אם יש לנו גישה לחשבון ה-AWS באמצעות aws cli:

```
PS > aws iam list-users
Unable to locate credentials. You can configure credentials by running "aws
configure".
```

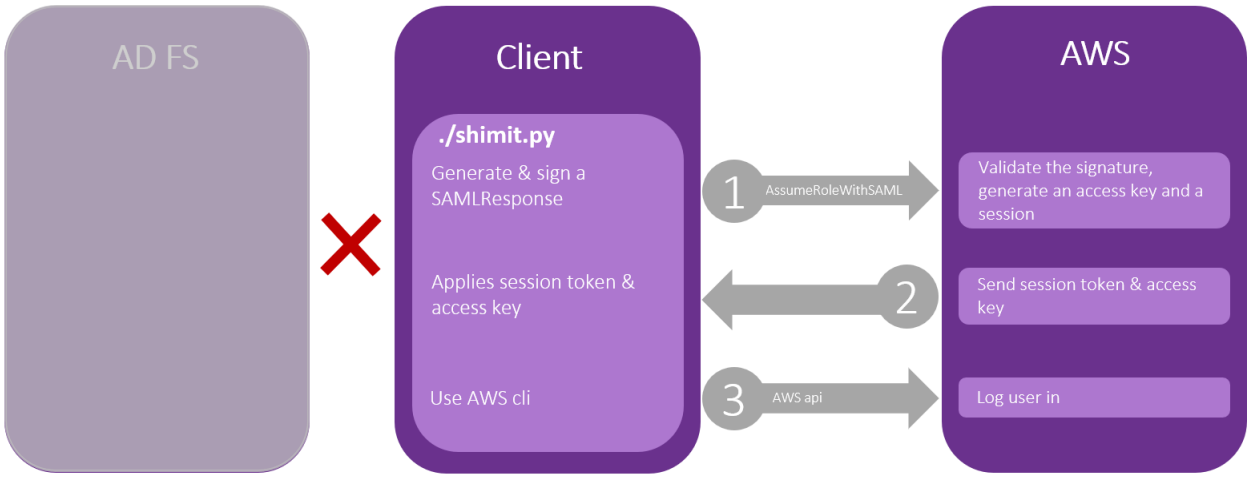
באופן לא מפתיע, אין לנו גישה לחשבון בשלב זה. נשתמש בכלי shimit על מנת לייצר SAMLResponse ולהתאמת בעזרתו אל חשבון ה-AWS:

```
PS > python .\shimit.py-idp http://adfs.lab.local/adfs/services/trust -pk key -c
cert.pem -u domain\admin -n admin@domain.com -r ADFS-admin -r ADFS-monitor -id
41[redacted]00

Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS > aws opsworks describe-my-user-profile
{
    "UserProfile": {
        "IamUserArn": "arn:aws:sts::[redacted]:assumed-role/ADFS Dev/admin@domain.com",
        "Name": "ADFS-Dev/admin@domain.com",
        "SshUsername": "adfs-dev-admindomaincom"
    }
}
```

דרך הפעולה של הכלי מתוארת בתרשים הבא:



1. ביצוע הזדהות מבוססת SAML:
   a. ייצור SAML Assertion המתאים לפרמטרים שסופקו על ידי המשתמש.
   b. חתימת ה-Assertion בעזרת המפתח הפרטי שנמצא בקובץ שסיפק המשתמש.
   c. פתיחת session מול ה-SP באמצעות AssumeRoleWithSAML() API ב-AWS.

2. קבלת Access Key ו-Session Token מ-AWS STS (השירות ב-AWS שמספק גישה זמנית ל-federated users).
3. שימוש בפרטי ההזדהות שהתקבלו על ידי שמירה שלהם במשתני סביבה בהם aws cli משתמש לצורך אימות מול השרת.

אף על פי שכל הפרטים הכתובים ב-SAMLResponse נמצאים בשליטתנו, ישנן מגבלות לטכניקה זו. אמנם ניתן לשלוט בפרמטר המציין מתי ה-SAMLResponse פג תוקף ולא ניתן להתאמת באמצעותו יותר (בעזרת הפרמטר SamlValidity), אך AWS בודק באופן מיוחד שאובייקט ה-SAMLResponse לא נוצר לפני יותר מ-5 דקות, בנוסף לבדיקה האם הוא עוד בתוקף.

סיכום

במאמר הצגנו איך תוקפים יכול להשתמש באחיזה ב-IdP של ארגון על מנת לקבל גישה מלאה לכל השירותים התומכים ב-SAML באותה פדרציה באמצעות Golden SAML. ראינו איך עקרון שיושם בעבר ליצירת Golden Ticket תקף גם לסביבות המבוססות על טכנולוגיות אחרות (ולא על Kerberos). היתרון הגדול של Golden SAML הוא היכולת של תוקף לקבל גישה לא מבוקרת לכל שירות בפדרציה (שתומך ב-SAML כמובן) הכוללת כל סט הרשאות שיבחר, ולשמור עליה לאורך זמן בצורה חשאית. אף על פי שישנה דרישת קדם לביצוע Golden SAML - השגת המפתח הפרטי של ה-IdP, טכניקה זו עדיין רלוונטית לתוקפים מעצמתיים למשל, מכיוון שאלה ירצו לבסס את אחיזתם, ולחיות בסביבה הנתקפת כמה שיותר זמן מבלי להתגלות גם אחרי שהשיגו גישה לנכסים החשובים ברשת.

לסיכום, נמנה מספר פעולות אותן יכולים מגנים לבצע על מנת למנוע/לזהות שימוש ב-Golden SAM:

1. הגנה על שרתי Identity Provider באותה הרמה שארגון מגן על שרתי ה-DC שלו, שכן שרתים אלה מספקים מידע על זהויות המשתמשים בארגון, בין אם ב-domain ובין אם בפדרציה.
2. ניהול הגישה למפתח הפרטי ולחשבון ה-ADFS כראוי. אופציה של החלפת המפתח המשמש לחתימה באופן תדיר יכולה גם היא להקשות על תוקפים בשימוש ב-Golden SAML, מכיוון שחתימה על SAMLResponse עם מפתח שהוחלף לא תאפשר לתוקף גישה לאף שירות. כמובן שאופציה זו דורשת מאמץ תשתיתי יותר גדול, שכן היא דורשת לעדכן את החלפת המפתח גם בכל השירותים הסומכים על אותו IdP.
3. ביצוע קורלציה בין רישום של התחברות SAML בצד ה-SP, לבין חתימת SAMLResponse בצד ה-IdP. במידה ונמצא רישום של התחברות באמצעות SAML ב-SP, אך אין כל רישום על ביצוע חתימה ב-IdP קודם לכן, ככל הנראה מדובר בשימוש ב-Golden SAML.

על המחבר

שקד ריינר, Security Researcher בחברת CyberArk. לכל שאלה, הערה או כל פניה אחרת ניתן ליצור קשר ShakedReiner@gmail.com.