עבודה מרחק - מתי נוחות הופכת למרחב תקיפה אידיאלי?

מאת דורין ראם

הקדמה

בעשור האחרון, עם הקורונה, המלחמות ושלל סיבות נוספות החלה עלייה משמעותית של עבודה מרחוק / מהבית (וזה גם דיי נוח אם נודה באמת). בעקבות המעבר הנרחב לעבודה מרחוק, ארגונים נדרשים לאפשר גישה מאובטחת למשאבי הרשת מחוץ לגבולות הפיזיים של הארגון. פתרונות VPN הפכו לאמצעי המרכזי לחיבור עובדים מרוחקים, אבל הם מספקים הגנה על תעבורת הנתונים בלבד ואינם מהווים מנגנון אימות זהות עצמאי. כאן, פרוטוקול RADIUS נכנס לתמונה.

מאמר זה בוחן את השימוש בפרוטוקול RADIUS כמנגנון אימות מרכזי בסביבות עבודה מרחוק, תוך ניתוח מנגנוני האימות השונים, אתגרי האבטחה הנלווים לשימוש בהם, דיון באיומים ומתקפות המבוססים על ניצול זהויות לגיטימיות. שאלת המחקר שלנו היא **כיצד הופכת התלות בפרוטוקול RADIUS לאימות עובדים מרחוק מאמצעי אמין למשטח תקיפה, ומהן ההשלכות האבטחתיות של מתקפות מבוססות זהות בסביבות Remote Access?**

רקע והיסטוריה

RADIUS (קיצור של Remote Authentication Dial-In User Service) - הפרוטוקול פותח לראשונה בשנת 1991 על ידי Livingston Enterprises במטרה לנהל חיבורי Dial-In לרשתות ארגוניות ולספק טכנולוגיית AAA למשתמשים. הפרוטוקול נועד להקל על מנהלי רשת בביצוע ניהול של חשבונות משתמשים ולספק רישום פעילות ברור של חיבורים מרוחקים.

עם הזמן, השימוש ב-RADIUS התרחב מעבר לחיבורי Dial-In קלאסיים והפך למרכיב מרכזי בניהול גישה מרחוק דרך VPN, Wi-Fi ארגוני (802.1X) ובקרות Network Access Control (NAC). כיום, RADIUS הינו גורם מרכזי בסביבות עבודה מרחוק, כל בקשת גישה עוברת דרכו ומנגנון האימות שלו הינו נקודת הגנה

קריטית מאוד בארגון.

קצת על Dial-In

Dial-In זה חיבור מרחוק לרשת הארגונית באמצעות טלפון ומודם. איך זה עבד?

- מחשב משתמש היה מתקשר למספר טלפון של השרת (RADIUS).
- המודם "צלצל" וקיבל חיבור קווי.
- המשתמש הזין שם משתמש וסיסמה, וקיבל גישה לרשת הפנימית של הארגון.

שירת את אותה המטרה, שעובדים היו צריכים להתחבר מהבית כדי לעבוד על שרתים ארגוניים. RADIUS נוצר במקור כדי לנהל את החיבורים האלה: מי מורשה להתחבר, מה הוא מורשה לעשות, ומה הוא עשה בזמן החיבור. המנגנון הזה, גם אם היה מיושן פיזית (טלפונים ומודמים), הוא סוג של האב הקדמון של האימות במערכות Remote Access מודרניות, כמו VPN.

ארכיטקטורת RADIUS

הפרוטוקול פועל במבנה Client-Server, ומחולק לשלושה גורמים עיקריים:

1. **RADIUS Client (ידוע גם כ-NAS):** יש כאן בלבול נפוץ, הלקוח הוא אינו המשתמש, אלא שרת הגישה (NAS). לדוגמא VPN Gateway או Access Point. התפקיד שלו זה בעצם להיות המתווך, הוא זה שלוקח את פרטי ההזדהות מהמשתמש, ושולח בקשת אימות לשרת RADIUS.
2. **RADIUS Server:** מבצע את תהליך האימות מול מערכות ניהול זהויות (LDAP ,AD), לאחר מכן מחזיר Access-Challenge / Access-Reject / Access-Accept. בנוסף, הוא מבצע גם רישום פעילות (Accounting) למעקב ובקרה. (מערכות NAC כיום פועלות בתפקיד זה).
3. **המשתמש / המכשיר המרוחק:** העובד שמנסה להתחבר באמצעות לפטופ, טלפון שמתחבר דרך WiFi וכו'.

המבנה הזה מאפשר ניהול מרכזי של אימות המשתמשים ובו זמנית שומר על אבטחה ככל הניתן, בנוסף הוא מספק יכולת מעקב ורישום פעילות. ההפרדה מסייעת לצמצם טעויות בהגדרות אבטחה ברכיבי הקצה ומקנה לארגון שליטה טובה יותר על הגישה לרשת.

פורמט ההודעות

RADIUS מבוסס UDP, עם פורטים נפוצים:

- 1812 - Authentication
- 1813 - Accounting

ההודעות הן AVPs - (Attributes / Values), שהם:

- User-Name
- User-Password
- NAS-IP-Address
- VLAN-Id
- Tunnel-Type

אופן הפעולה של RADIUS עם VPN ועקרון AAA

בתרחיש עבודה מרחוק, המשתמש המרוחק (למשל עובד מהבית) מנסה לגשת למשאבים פנימיים של הארגון באמצעות VPN , שמספק חיבור מוצפן בין המכשיר שלו לרשת הארגונית. בקשת הגישה הזו עוברת דרך RADIUS, שמיישם את עקרון ה-AAA בצורה מסודרת.

1. שלב בקשת האימות - Authentication (Access-Request):

המשתמש מכניס את פרטי האימות, או מנגנוני אימות מתקדמים יותר כמו EAP-TLS, PEAP או MFA (עליהם אפרט בהמשך). ה-NAS - RADIUS Client, לוקח את הנתונים ושולח בקשת Access-Request לשרת RADIUS. הבקשה כוללת את שם המשתמש, הסיסמה או Credential אחר, כתובת ה-IP של המשתמש, וסוג השירות המבוקש (VPN Wi-Fi וכו'). שרת ה-RADIUS (שמהווה חלק ממערכת ה-NAC) מחבר את הבקשה למקור האימות ובודק אם המשתמש קיים, האם הסיסמה נכונה, והאם למשתמש יש הרשאות מתאימות לפי מדיניות הארגון.

בשלב זה בעצם נשאלת השאלה: מי המשתמש?

2. שלב החלטת הגישה (Authorization) Access-Challenge / Access-Reject / Access-Accept

לאחר אימות הזהות, שרת הרדיוס מחליט אילו משאבים ומדיניות גישה הלקוח זכאי להם.

a. Access-Accept: המשתמש מורשה להתחבר.

b. Access-Reject: המשתמש לא מורשה.

c. Access-Challenge: השרת דורש מידע נוסף לפני מתן גישה, לדוגמה OTP.

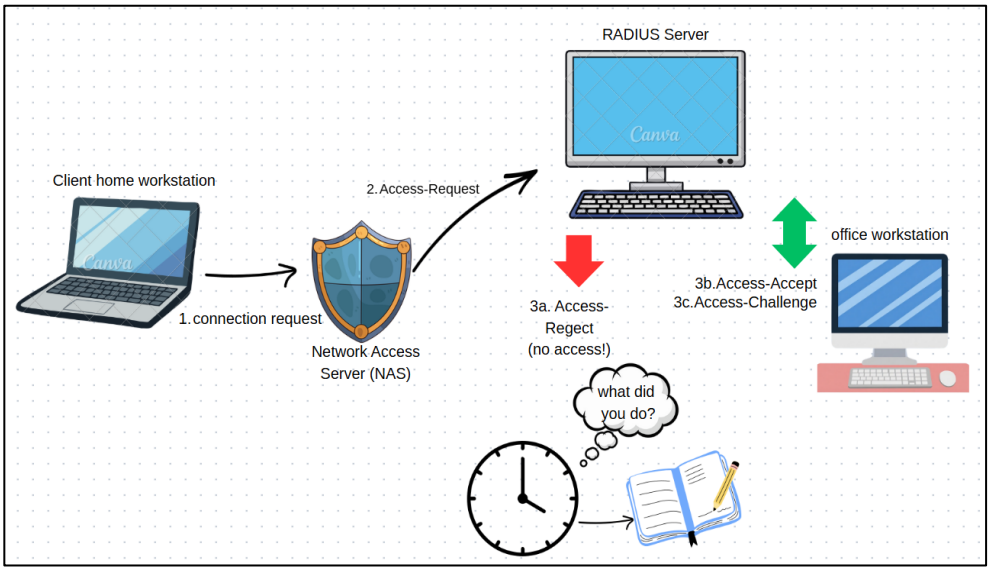
בשלב זה בעצם נשאלת השאלה: מה מותר למשתמש הזה לעשות?

3. שלב הרישום Accounting

במקביל, השרת רדיוס רושם את פרטי החיבור: מיהו המשתמש, מתי התחבר, משך החיבור, כתובת ה-IP שהוקצתה וסוג השירות שהשתמש בו. בקיצור, כל מידע שיכול להיות חיוני למעקב, audit ולזיהוי פעילות חשודה ברשת.

בשלב זה בעצם נתעד: מה המשתמש הזה עשה?

לאחר שלב AAA, השרת מחזיר ל-VPN Gateway החלטה מתאימה: Access-Accept במידה והמשתמש מורשה, Access-Reject אם לא, או Access-Challenge אם נדרש מידע נוסף (למשל OTP). הלקוח מאפשר את החיבור למשתמש במידה ואושר, וכל התעבורה לאחר מכן מוגנת בהצפנה.



נקודות אבטחה קריטיות

- כל החלטת גישה מבוססת על אימות זהות, ולכן כל ניצול של Credentials לגיטימיים מהווה סיכון משמעותי.
- מנגנוני Authorization יכולים להגביל נזקים במקרה של גישה לא מורשית.
- רישום ה-Accounting מאפשר גילוי של פעילות חריגה ומעקב אחרי דברים שנראים פחות לגיטימיים ברשת.

הסבר כללי על הפרוטוקול בשילוב של RADIUS

פרוטוקול אימות שפותח על ידי מייקרוסופט ונעשה בו שימוש גדול במערכות גישה מרחוק, בין השאר בסביבות VPN ובמערכות אימות מבוססות RADIUS, כפי שמוגדר ב-RFC2759. כמו שאמרנו מקודם, שרת ה-RADIUS מתפקד כגורם אימות מרכזי ומתווך בין לקוח הקצה לבין שירותי הגישה לרשת. עכשיו כאן נוסף לנו MS-CHAPv2, כדי לשמש כתוספת למנגנון האימות.

התהליך המשולב שלהם הולך כך: המשתמש פונה ל-VPN, שרת הרדיוס מייצר ערך אקראי (challenge) ושולח ללקוח. כאן נכנס באמת התפקיד של MS-CHAPv2, הלקוח ייקח את ה-NT-Hash (מבוסס MD4) של הסיסמא שלו ומבצע איתו את החישוב על ה-challenge שהתקבל. את התשובה הוא מחזיר לשרת. בינתיים בצד, השרת מבצע את אותו החישוב בעזרת שליפה של ה-hash של הסיסמא מה-AD, וביצוע של החישוב על אותו ה-challenge בעזרתו. אם התוצאות זהות תישלח הודעת Access-Accept. אם שונות תישלח הודעת Access-Reject.

יתרונות אבטחתיים

אחד היתרונות האבטחתיים המרכזיים של MS-CHAPv2 הוא הימנעות מהעברת הסיסמה של המשתמש באופן גלוי במהלך תהליך האימות - הרי הפרוטוקול משתמש במנגנון קלאסי של Challenge-Response, שמבטיח כי המידע המועבר ברשת הוא תוצר של חישוב קריפטוגרפי ולא הסיסמה עצמה. בנוסף, הוא משתמש ב-NT-Hash ולא בסיסמה עצמה.

יתרון נוסף הוא התמיכה באימות דו-כיווני (Mutual Authentication), המאפשר גם ללקוח לאמת את זהות השרת, ובכך תורם להפחתת הסיכון להתקפות של התחזות לשרת. הפרוטוקול משתמש בערכי challenge שונים לכל session, מה שמקטין את האפשרות לשימוש חוזר בנתוני אימות שנלכדו בעבר ומסייעים בהגנה מפני התקפות Replay.

בהקשר של RADIUS, שילוב MS-CHAPv2 מאפשר ריכוז תהליכי אימות וניהול זהויות בשרת מרכזי, תוך כדי גם אכיפת מדיניות אבטחה אחידה ושילוב עם Active Directory, ללא צורך באחסון סיסמאות באופן גלוי (אלא NT-Hash כמו שאמרנו קודם) על רכיבים ברשת.

יתרון נוסף הוא ש-MS-CHAPv2 נחשב לפרוטוקול קל ומהיר ליישום, במיוחד בסביבות מבוססות מיקרוסופט, שכן הוא נתמך באופן מובנה במערכות הפעלה, שרתי RADIUS ופתרונות VPN נפוצים, וזה פשוט מאפשר פריסה מהירה תחזוקה נוחה ושמירה על מנגנון אימות מובנה ומוכר לארגון.

עד כאן - נשמע טוב, לא? אבל תכלס, האבטחה בפועל נשענת על אלגוריתמים קריפטוגרפים מיושנים!

אז מה הבעיה?

למרות ש-MS-CHAPv2 מיישם עקרונות נכונים של אימות מבוסס אתגר-תגובה ואימות דו-כיווני, הפרוטוקול סובל מחולשות קריפטוגרפיות שהופכות אותו לפגיע למתקפות יירוט ופיצוח (/ Offline (Eavesdropping Interception Attacks, מתקפה פסיבית שבה תוקף מאזין לתקשורת בין שני צדדים ברשת לצורך איסוף מידע רגיש, ללא שינוי או שיבוש התקשורת עצמה - במקרה שלנו, המתקפה מתמקדת בלכידת הודעות האימות (כגון ערכי אתגר ותגובות), שלמרות שאינן כוללות את הסיסמה עצמה, הן עשויות להכיל מידע מספיק לצורך ניתוח קריפטוגרפי ופיצוח זהות המשתמש. לכן, למרות שהפרוטוקול אינו מעביר את סיסמת המשתמש בטקסט גלוי, תוקף בעל גישה לערוץ התקשורת יכול להשיג את הודעות האימות (ערך אתגר ותגובת הלקוח) ולבצע מתקפת Brute Force או פיצוח Offline של סיסמאות, ללא אינטראקציה נוספת עם השרת. החולשה נובעת מהסתמכות על אלגוריתם DES עם מפתח אפקטיבי באורך 56 ביט, דבר שמצמצם את מרחב החיפוש של הסיסמה ומאפשר פיצוח מהיר יחסית.

החולשות האלו זוהו ונותחו לאורך מספר שנים על ידי חוקרי אבטחת מידע. כבר בסוף שנות ה-90 הצביעו חוקרים על בעיות מהותיות במנגנוני האימות של PPTP והרחבותיו, ובפרט על MS-CHAP, תוך ניתוח מבני של השימוש באלגוריתמים קריפטוגרפים חלשים. בתחילת שנות ה-2000 פורסמו עבודות מחקר נוספות שהעמיקו בניתוח הפרוטוקול והדגימו חולשות נוספות במימוש ובמודל האיום שלו. בשנת 2012 כבר הראו כי ניתן לצמצם את אבטחת MS-CHAPv2 לבעיית פיצוח של מפתח DES יחיד, ובכך הדגימו מתקפת פיצוח יעילה על תהליך האימות כולו. ממצאים אלו חיזקו עוד את ההוכחה ש-MS-CHAPv2 אינו מאובטח כל כך, והובילו להמלצות להימנע משימוש בו לבד.

סיכום חולשות

אז בקצרה כמו שאמרנו, החולשות המרכזיות של MS-CHAPv2 נובעות מהצפנה חלשה והסתמכות על אלגוריתמים מיושנים כמו NT-Hash ו-DES, אשר מאפשרים לפצח סיסמאות שנלכדו בתהליך האימות באמצעות מתקפות Brute-Force או Rainbow Tables. הפרוטוקול אינו מספק הגנה חזקה מול מתקפות Man-in-the-Middle או Credential Relay, מה שמאפשר לתוקף להשתמש בנתוני האימות גם ללא ידיעת הסיסמה עצמה.

לכן, MS-CHAPv2 כמעט ואינו משמש כיום כפרוטוקול עצמאי, אלא נעטף בתוך ערוצי הצפנה חזקים כמו PEAP או TLS ולעיתים משולב עם מנגנוני MFA, כדי להבטיח אבטחה נוספת. בסביבות Remote Access, האיום אינו מוגבל למחשבים בתוך הארגון: רשת WiFi עוינת, VPN עם קונפיגורציה חלשה או מכשיר קצה שנפרץ יכולים להספיק כדי להשיג את הפרטים הרלוונטיים. אם לתוקף יש את תהליך האימות, הוא יכול להתחבר כמשתמש לגיטימי, שכן שרת ה-RADIUS יאשר את הגישה. MS-CHAPv2 נחשב נפוץ לתקיפות של Identity Abuse ו-Living off the Land, תקיפות שבהן התוקף משתמש בהרשאות קיימות ובכלים

לגיטימיים כדי לנוע בתוך המערכת מבלי להפעיל מנגנוני גילוי מסורתיים.

הסבר קצר על Living off the Land:

Living off the Land הינה תקיפה שבה התוקף לא מכניס כלים זדוניים חדשים למערכת, אלא עושה שימוש בכלים, מנגנונים וזהויות לגיטימיים הקיימים כבר בסביבה הארגונית. כלומר אין שימוש ב-malware של ממש, לא קוד זדוני וכו'. אלא בפעולה באמצעות הרשאות קיימות וזהות תקפה, דבר המקשה על זיהוי הפעילות כזדונית. בתוך הקשר זה, **Identity Abuse** נחשב לתת-קטגוריה, הוא מתבסס על שימוש לרעה בזהויות לגיטימיות כגון שם משתמש וסיסמה, טוקן VPN, תעודת משתמש או Session פעיל. מאחר והפעילות מתבצעת תוך שימוש באמצעי אימות והרשאות תקינים, מערכות האבטחה מפרשות את הגישה כלגיטימית, ה-SOC עלול לא לשים לב שמשהו בכלל קורה, והתוקף מסוגל לפעול במערכת מבלי לעורר התראות חריגות. לכן, קשה מאוד לדעת כשדבר כזה קורה, קשה לניטור וקשה לטיפול. בקצרה, המתקפות משתמשות בזהויות לגיטימיות כדי לעקוף מנגנוני אבטחה. בסביבות עבודה מרחוק המבוססות על RADIUS ו-VPN, המתקפות האלו מהוות איום משמעותי, שכן מנגנון האימות מאשר גישה על סמך זהות בלבד, ללא יכולת מובנית להבחין בין משתמש לגיטימי לתוקף המחזיק בפרטי הזדהות תקפים.

למה זה רלוונטי? כי זה עדיין משומש. והרבה.

- Legacy Systems - מערכות ישנות שלא תומכות ב-EAP-TLS (שתכף נפרט עליו)
- נוחות תפעולית - אין תעודות, אין PKI
- עטיפה ב-PEAP (אסביר בהמשך גם)

על אף הפגיעויות הקריפטוגרפיות הידועות של MS-CHAPv2, הפרוטוקול עדיין נמצא בשימוש נרחב בסביבות ארגוניות, בעיקר בשל תאימות לאחור ונוחות תפעולית. המצב הזה מדגיש פער משמעותי בין תפיסת האבטחה של ארגונים לבין רמת ההגנה בפועל, במיוחד בסביבות עבודה מרחוק המבוססות על זהויות לגיטימיות.

טוב, אז יש בעיה אבטחתית.. ויש גם פתרון פלסטר עבורה. רגע לפני שנצלול אליו, בואו נדבר רגע על הקונספט של EAP.

EAP - Extensible Authentication Protocol

EAP הוא פרוטוקול מסגרת (Framework) לאימות זהויות ברשת.

הוא לא מנגנון אימות בפני עצמו, אלא מאפשר "להצמיד" מנגנוני אימות שונים:

- Passwords (MS-CHAPv2)
- Certificates (EAP-TLS)
- One-Time Passwords (EAP-MFA / OTP)

איך זה עובד בקצרה

- הלקוח שולח **EAP Request** לשרת.
- השרת מחזיר **EAP Response**.
- התקשורת הזו יכולה להיות פשוטה (Password) או מורכבת (TLS, Token, MFA).
- הפרוטוקול מאפשר **Negotiation** - הלקוח והשרת מסכימים איזה מנגנון אימות להשתמש.

למה EAP חשוב ב-RADIUS

- RADIUS "מסתכל" על EAP כעל מנגנון Authentication Plug-and-Play.
- הוא מאפשר החלפה קלה של מנגנון אימות, בלי לשנות את ה-RADIUS Server או ה-VPN Gateway.

PEAP = Protected EAP

- PEAP הוא בעצם EAP עטוף בערוץ TLS. כלומר, ה-EAP Framework נשאר, אבל כל מנגנוני האימות שמריצים בתוך EAP (למשל MS-CHAPv2 או סיסמה רגילה) מוגנים באמצעות TLS מוצפן.
- במילים אחרות: EAP = המסגרת, PEAP = הגנה על המסגרת.

הסבר קצר על תכלס איך PEAP עובד ב-VPN + RADIUS

1. המשתמש מנסה להתחבר ל-**RADIUS Client** = VPN → VPN Gateway
2. RADIUS Server שולח **EAP Request בתוך ערוץ** → **TLS** נוצר tunnel מוצפן
3. בתוך ה-tunnel:
    - מבוצע מנגנון האימות שבחרו (למשל MS-CHAPv2)
    - ה-Credentials מוגנים מפני האזנה ברשת
4. Access-Accept / Access-Reject / Access-Challenge
5. Accounting ברישום פרטי החיבור

יתרונות PEAP

- מונע **sniffing** ו-MITM על תעבורת האימות.
- מאפשר שימוש ב-MS-CHAPv2.
- ניתן לשלב עם OTP.
- כמעט תמיד תואם לכל VPN / NAS.

מגבלות

- TLS עצמו חייב להיות מוגדר טוב.
- אם המנגנון הפנימי חלש - עדיין אפשר brute-force בתוך הערוץ המוצפן.
- **לא מונע Identity Abuse** - אם Credentials נגנבים, התוקף עדיין יכול להתחבר לגיטימי.

לסיכום, PEAP מספק שכבת הגנה קריטית על מנגנוני אימות חלשים, על ידי הצפנת כל תהליך האימות בערוץ TLS, אבל אינו מבטל את הצורך במנגנוני אימות חזקים ומעקב אחר שימוש זהויות לגיטימיות בסביבות עבודה מרחוק.

EAP-TLS - אימות מבוסס תעודות

מה זה EAP-TLS?

EAP-TLS הוא מנגנון אימות בתוך EAP שמבוסס על תעודות דיגיטליות (Certificates), כלומר - אין שימוש בסיסמאות בכלל. האימות מתבצע באמצעות קריפטוגרפיה אסימטרית, לא סיסמאות.

איך זה עובד ב-VPN + RADIUS?

1. המשתמש מנסה להתחבר ל-VPN.
2. מתחיל TLS Handshake בין הלקוח לשרת RADIUS.
3. שני הצדדים מזדהים באמצעות תעודות - השרת מציג תעודה (Server Certificate) והלקוח מציג תעודה (Client Certificate).
4. השרת בודק: האם התעודה של הלקוח חתומה ע"י CA מהימן, האם היא בתוקף והאם היא שייכת למשתמש/מכשיר מורשה.
5. אם הכול תקין - Access-Accept. אם לא - Access-Reject
   - אין Challenge-Response
   - אין סיסמה.
   - אין OTP (אם כי אפשר לשלב).

למה זה הרבה יותר מאובטח?

- אין סיסמאות לגניבה
- חסין ל-Phishing
- חסין ל-MITM
- חסין ל-Brute Force
- קשה מאוד לבצע Identity Abuse

ככה שגם אם תוקף מאזין לרשת, משיג תעבורה ורואה את כל ה-TLS.. הוא לא יכול להתחבר בלי המפתח הפרטי של התעודה, שנשמר על המכשיר.

אז למה כולם לא משתמשים בזה?

אז זה לא כל כך פשוט. כאילו באמת - זה לא פשוט! על אף ש-EAP-TLS נחשב למנגנון האימות המאובטח ביותר בסביבות RADIUS, שימוש בו לא נפוץ בארגונים. הסיבה המרכזית לכך היא אינה טכנולוגית אלא תפעולית: EAP-TLS מחייב הקמה ותחזוקה של תשתית PKI (תכף ארחיב) מלאה, הכוללת הנפקה, ניהול וביטול תעודות דיגיטליות עבור משתמשים ומכשירים. תהליך זה מעלה את רמת המורכבות התפעולית בכמה רמות, דורש ידע מקצועי ומשאבים (יקר..), ומקשה במיוחד בסביבות הכוללות מכשירים לא מנוהלים או משתמשים זמניים. לכן, ארגונים רבים מעדיפים פתרונות נוחים יותר כגון PEAP, גם במחיר של ויתור על רמת אבטחה גבוהה יותר.

PKI בקצרה - ההקשר ל-EAP-TLS

PKI - Public Key Infrastructure - היא תשתית לניהול תעודות דיגיטליות, המאפשרת אימות זהות מבוסס קריפטוגרפיה אסימטרית ללא שימוש בסיסמאות. במסגרת זו, כל משתמש או מכשיר מחזיק בתעודה דיגיטלית החתומה על ידי גורם מהימן (CA), המוכיחה את זהותו באמצעות מפתח פרטי הנשמר בצד הלקוח. ב-EAP-TLS, אימות הזהות מתבצע באמצעות בדיקה הדדית של תעודות, מה שמספק רמת אבטחה גבוהה במיוחד אך דורש ניהול מחזור חיים מלא של התעודות.

Shared Secret כצוואר בקבוק

חשוב להבין לגבי ה-shared secret שגם כאן ישנה חולשה כלשהי. החולשה לא תלויה בפרוטוקול האימות עצמו, אלא במנגנון האמון בין שרת ה-RADIUS ל-NAS. כל פרוטוקול אימות שמיושם דרך RADIUS - בין אם זה MS-CHAPv2 או EAP או אחרים - מסתמך על shared secret משותף כדי לאמת את ההודעות ולצמצם זיוף. כתוצאה מכך, אם הסוד חלש, קצר, דלף או קשה לתפעול, כל מנגנון האימות הופך לפגיע להתקפות כגון Replay או Message Forgery.

כלומר, הבעיה היא ארכיטקטונית וגורפת לכל פרוטוקול אימות, ולא מוגבלת ל-MS-CHAPv2 בלבד (שנחשב חלש בלי קשר), אך פרוטוקולים חלשים כמו MS-CHAPv2 מגדילים את הסיכון לניצול החולשה בפועל.

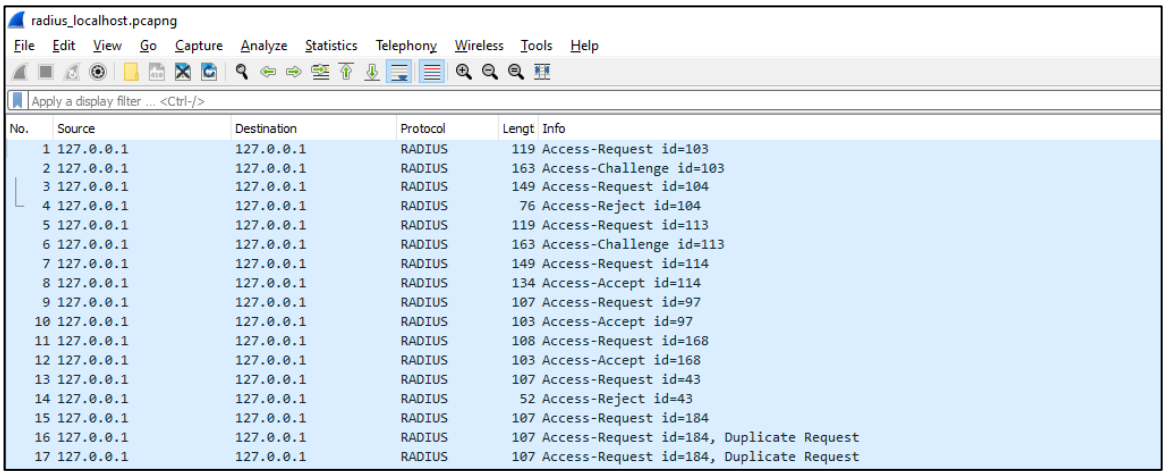
אבטחה מובנית מול אבטחה בפועל

השימוש הנפוץ ב-PEAP כמנגנון אימות בסביבות RADIUS ובעבודה מרחוק מספק הצפנה של תהליך האימות באמצעות TLS, אך ממשיך להסתמך על סיסמאות כבסיס לזיהוי המשתמש. תלות זו חושפת את הארגון למתקפות המבוססות על גניבת פרטי הזדהות, כגון פישינג או שימוש חוזר בסיסמאות, שבהן תוקף יכול להשיג גישה לגיטימית לרשת הארגונית ללא צורך בניצול חולשות טכניות.

אימות מבוסס תעודות באמצעות EAP-TLS מציע רמת אבטחה גבוהה יותר, שכן הוא מבטל את השימוש בסיסמאות וקושר את הזהות למכשיר באמצעות מפתח פרטי. עם זאת, הדרישה לניהול תשתית PKI מעלה את מורכבות התפעול, ומהווה גורם מרכזי לכך שארגונים רבים ממשיכים להשתמש ב-PEAP, גם במחיר של סיכון אבטחתי מוגבר.

קצת תקשורת

כדי להבין איך הפרוטוקול ממש מתנהג בפועל ברמת הרשת ואיך זה נראה, נסתכל על דוגמא של PCAP מתוך מאגר דוגמאות התקשורת של Wireshark. ב-PCAP ניתן לראות שהתקשורת מתבצעת בסביבה מקומית (127.0.0.1) ואת הביצוע של המנגנונים המרכזיים בעבודה מול שרת רדיוס:



1. אימות מבוסס אתגר (challenge response): בפקטות 1-8 נראה שהשרת לא מאשר את ההתחברות מיד אלא מחזיר הודעת Access-challenge. זה קורה בשיטות אימות MS-CHAPv2 או EAP, על מנת לתת עוד מגנון הגנה. כפי שהסברתי ניתן לראות שניסיון אחד מסתיים בדחייה (Access-Reject) ואחד מסתיים בהצלחה (Access-Accept).
2. אימות ישיר: בפקטות 9-14 התהליך קצר הרבה יותר, הלקוח שולח בקשת Access-Request והשרת מחזיר תשובה מיידית של אישור או דחייה, ללא שלב אתגר באמצע (אופייני לשיטות כמו PAP).
3. התמודדות עם איבוד חבילות: בחבילות 15-17 מופיעה התנהגות שבה הלקוח שולח בקשה ולא מקבל תשובה מהשרת בזמן ולכן מנגנון ה-retry שלו שולח את הבקשה שוב ושוב (מה שאנחנו רואים כ-Duplicate request.

תקיפות נפוצות

אז אחרי שדיברנו בכללי על איך רדיוס עובד, נדבר על כמה מתקפות ידועות שמדגימות לנו את הבעיות שבפרוטוקול.

Evil Twin

בתרחיש Evil Twin, התוקף מקים נקודת גישה (AP) זדונית בעלת SSID זהה לרשת הארגונית ומפעיל שרת RADIUS מזויף. בבסיס המתקפה עומד ניצול של פרוטוקול X802.1, כאשר ה-AP משמש כ-Authenticator והשרת המזויף כ-Authentication Server. אם תחנת הקצה אינה מוגדרת לאמת את תעודת ה-SSL/TLS של השרת (Server Validation), היא תשלים את תהליך האימות מול ישות עוינת.

מנגנון חשיפת פרטי ההזדהות

בפרוטוקולים נפוצים כגון EAP-PEAP או EAP-TTLS, מוקם בשלב הראשון ערוץ TLS המבוסס על אימות צד-שרת בלבד. בתוך הערוץ ה"מאובטח" לכאורה, מבוצע אימות פנימי (לרוב בתצורת MS-CHAPv2). בשלב זה, הלקוח משיב ל-Challenge של השרת בתגובה קריפטוגרפית הנגזרת מסיסמת המשתמש. התוקף, השולט בשרת ה-RADIUS, לוכד את ה-Challenge/Response ומבצע Offline Cracking של ה-NTLM Hash, ללא צורך באינטראקציה נוספת עם הקורבן.

מעבר רוחבי (Lateral Movement) ל-VPN

הסיכון האמיתי מתממש כאשר אותם פרטי הזדהות (Credentials) משמשים גם לגישה מרחוק. בתצורות רבות, שרתי VPN (בין אם SSL או IPsec) מבצעים אימות מול אותו מסד נתונים ארגוני באמצעות RADIUS. הצלחת התקיפה ברובד ה-WiFi מעניקה לתוקף "מפתח מאסטר":

- התוקף יכול להתחבר ל-VPN הארגוני מכל מקום בעולם.
- התקיפה עוקפת בקרות היקפיות כגון Firewall או מערכות NAC המוגבלות לרשת המקומית.
- זיהוי התקיפה הופך לקשה במיוחד, שכן החיבור נראה לגיטימי לחלוטין במערכות הניטור.

סיכום: כשל ארכיטקטוני

המתקפה מדגימה כשל תכנוני עמוק ולא רק טעות הגדרה מקומית. הסתמכות על אימות מבוסס סיסמה ב-RADIUS, ללא אכיפת אימות הדדי (Mutual Authentication) מבוסס תעודות (כגון EAP-TLS) וללא הפרדה סגמנטלית בין הרשאות הגישה לרשת האלחוטית לגישת ה-VPN, מייצרת נקודת כשל יחידה (Single Point of Failure). זה שאין MFA (אימות רב-שלבי) בנקודות הקצה הללו הופך את הארגון לחשוף למעבר רוחבי מהיר בין תשתיות גישה שונות.

Blast-RADIUS - CVE-2024-3596

מתקפת blast-radius שנחשפה ב-2024 היא פגיעות ברמת הפרוטוקול, ולא במימוש כלשהו. היא ממש תוקפת את הפרוטוקול במשהו שהוגדר ב-RFC 2865 ומנצלת חולשה קריפטוגרפית בשימוש שהוא עושה ב-MD5.

מנגנון הכשל הקריפטוגרפי

מקור הבעיה הוא בשדה ה-Response Authenticator. השדה הזה אמור להוכיח ללקוח ה-RADIUS (ה-NAS) שהתשובה הגיעה מהשרת המורשה (RADIUS/NAC) ושהיא לא עברה שינוי. החישוב של השדה מתבצע על ידי הרצת MD5 על שילוב של קוד ההודעה, המזהה (Identifier), אורך החבילה, ה-Request Authenticator, ה-Attributes והסוד המשותף (Shared Secret).

המתקפה מנצלת את העובדה ש-MD5 פגיע ל-Chosen-Prefix Collision. תוקף שנמצא בעמדת MitM (בין ה-NAS לשרת ה-RADIUS) יכול להכניס מה שהוא רוצה למאפיין proxy-state בבקשת ה-Access-Request המקורית. בזכות החולשה של MD5, התוקף מסוגל לייצר חבילת Access-Accept מזויפת שתהיה לה את אותו Hash כמו לחבילת ה-Access-Reject האמיתית ששלח השרת. לאחר מכן יוצא מצב שבו ה-NAS מקבל את החבילה המזויפת, מוודא את החתימה מול הסוד המשותף (שנשארת תקינה למרות הזיוף), ומאשר כניסה למשתמש לא מורשה, והתוקף בכלל לא צריך לדעת את ה-Shared Secret עצמו.

קצת יותר על Chosen-Prefix Collision

במתקפת Collision המטרה היא למצוא שני קבצים שונים שנותנים את אותו ה-Hash. ב-Chosen-Prefix Collision, התוקף בוחר שתי התחלות שונות (Prefixes), ואז הוא מחשב "זנב" (Suffix) לכל אחד מהם, כך שהתוצאה הסופית של שניהם תהיה זהה ב-Hash. בהודעה הראשונה בעצם ישנה דחייה - פה התוקף יוסיף משהו, ובהודעה השנייה יש אישור - וגם פה הוא יוסיף אחרי משהו. בסופו של דבר - לשתיהן יהיה אותו hash. אבל מה הוא מוסיף ואיך לשתיהן?

התוקף יבחר להזריק את הג'יבריש שלו - ל-Proxy-State, מאפיין שקיים בהודעות (גם מהלקוח לשרת וגם ההפך). בפרוטוקול רדיוס מוגדר שאם לקוח שולח הודעה עם שדה שנקרא proxy-state, השרת חייב להעתיק את השדה הזה כפי שהוא לתשובה שלו. לכן התוקף בעצם יודע שגם אם הוא לא יכול להכריח את השרת לתת לו "אישור "במקום "דחייה", יש משהו שהוא יכול להכריח אותו לתת. בקיצור - הוא מזריק את הג'יבריש להודעת הבקשה המקורית. לאחר מכן - השרת מקבל את הבקשה, רואה שהסיסמה לא תואמת ויוצר הודעת דחייה, הוא מעתיק להודעה הזאת את ה-proxy-state שנתן התוקף, וחותם על החבילה. עכשיו מגיע שלב ההחלפה - התוקף תופס את ההודעה החתומה (הרי הוא בין השרת ללקוח), עכשיו יש לו ביד חתימה חוקית של השרת שמתאימה מתמטית גם להודעת ה-Accept שהכין מראש. הוא שולח ללקוח (ה-VPN) את ההודעה שהוא יצר בעצמו, ה-accept, ובגלל שהחתימה נכונה - הוא מקבל את מבוקשו.

עבור שרתי VPN הדבר הזה ממש קריטי. רוב מערכות ה-VPN (כמו IPsec או SSL-VPN) מסתמכות על RADIUS לאישור הרשאות.

מה שקורה בקצרה הוא:

1. המשתמש (התוקף) מנסה להתחבר עם פרטים שגויים.
2. שרת ה-VPN שולח בקשה ל-RADIUS.
3. השרת מחזיר דחייה (Access-Reject).
4. התוקף מבצע את ה-Collision בזמן אמת (כל ההסבר מקודם), מחליף את תוכן החבילה ל-Accept, ושומר על חתימה זהה.
5. ה-VPN "משתכנע" שהאימות הצליח ופותח את ה-Tunnel לרשת הארגונית.

הסיכון גבוה במיוחד מאחר שתעבורת RADIUS עוברת לרוב ב-UDP מעל פורט 1812 ללא הצפנה נוספת, מה שמאפשר לתוקף שנמצא בנתיב התקשורת לבצע מניפולציות על חבילות בקלות יחסית.

העתיד של הגישה המרוחקת: מ-VPN ו-RADIUS לעבר ZTNA

פרוטוקול RADIUS ופתרונות ה-VPN עובדים כך שברגע שמשתמש אימת את עצמו הוא יכול לנוע ברשת, אבל כיום זה נחשב כבר גישה לא לגמרי מאובטחת. עולם אבטחת המידע צועד לעבר תפיסת ה-**Zero Trust Network Access**. השאלה המרכזית היא האם ZTNA עתיד להחליף לחלוטין את התשתיות הקיימות.

ההבדל המהותי טמון במעבר מאימות נקודתי לניטור רציף. בעוד ש-RADIUS מבצע אימות חד פעמי ברגע החיבור (ומסתמך לעיתים על פרוטוקולים חלשים כמו MS-CHAPv2), ZTNA פועל לפי העיקרון של "לעולם אל תסמוך, תמיד תאמת". בגישה זו, הגישה אינה ניתנת לכל הרשת הארגונית, אלא באופן ספציפי לאפליקציה או למשאב הנדרש בלבד (Least Privilege).

האם RADIUS יעלם? סביר להניח שלא בעתיד הקרוב. ארגונים רבים מחזיקים בתשתיות Legacy (מערכות ישנות) שעדיין דורשות RADIUS לצורך ניהול גישה לציוד תקשורת גם פיזית וגם רשתות Wi-Fi ארגוניות. למרות זאת, עבור עבודה מרחוק, ה-VPN הופך באמת פשוט לנטל אבטחתי. ZTNA פותר הרבה מהבעיות שהצגנו במחקר: הוא מבטל את משטח התקיפה הגלוי (ה-VPN Gateway אינו חשוף לאינטרנט באותה צורה), והוא דורש אימות חזק ורציף שאינו מתבסס רק על שם משתמש וסיסמה שנשלחים ב-UDP. המעבר ל-ZTNA מייצג שינוי פרדיגמה: במקום להתמקד ב"איך המשתמש מתחבר", מתמקדים ב"מה המשתמש עושה" ובאימות הזהות והקשר המכשיר (Device Context) בכל רגע נתון. עבור ארגונים המבקשים להתגונן מפני מתקפות מורכבות כמו Blast-RADIUS או גניבת זהויות לגיטימיות, הדבר רלוונטי מאוד.

שיטות הגנה והמלצות

אז מה אנחנו יכולים לעשות? קודם כל כמובן - להיות מודעים לסיטואציה. הנושא הזה לא מדובר כל כך הרבה, למרות שהוא מאוד רלוונטי.

להפחית את הסיכונים הכרוכים ב-MS-CHAPv2 ובמערכות RADIUS, לא להסתמך על הפרוטוקול כפתרון עצמאי, אלא לשלבו בתוך מסגרות הצפנה מתקדמות כגון PEAP או TLS, ולהוסיף שכבות הגנה נוספות כמו MFA - Multi-Factor Authentication, שמקטינות את הסיכון לניצול Credentials שנגנבו.

מומלץ להשתמש ב-Shared Secret חזק, ארוך ומורכב, וליישם מדיניות ניהול סודות שמקלה על סנכרון ושינוי תקופתי כדי למנוע חשיפה.

להחליף סיסמאות פעם בפרק זמן לא גדול מדי (ותמיד לסיסמה מאובטחת ומסובכת).

בנוסף, מומלץ לאכוף שימוש בערוצי תקשורת מוצפנים בכל שלב של Remote Access, כולל VPN ורשתות Wi-Fi ארגוניות, ולהגביל הרשאות למינימום הנדרש כדי למנוע ניצול Identity Abuse. חשוב לשלב מנגנוני ניטור וזיהוי כדי לזהות פעילות לא רגילה של משתמשים לגיטימיים, שכן מתקפות מסוג Living off the Land מסתמכות על כלים קיימים והן קשות לזיהוי באמצעים מסורתיים. בקיצור - לנטר גם על משתמשים לגיטימיים ולשים לב למה שהם עושים, מאיזה רכיבים הם מתחברים.

לבסוף, שינוי סיסמאות תקופתי, ניהול זהויות והדרכת משתמשים לגבי סיסמאות חזקה.

סיכום

הבנו שכולנו אוהבים את הקונספט של לעבוד מהבית (לפחות מדי פעם) ובו זמנית אנחנו אנשים ש(אני מניחה) אכפת להם מרמת האבטחה בחברה שבה אנחנו עובדים.

הכרנו את פרוטוקול RADIUS - הרקע שלו, המבנה, ואיך הוא פועל. דיברנו על מנגנוני האימות הנפוצים בשימוש בו, מה היתרונות לעומת החסרונות של כל אחד מהם. בנוסף עברנו על וקטורי תקיפה לדוגמא, ובחנו את רמות הסיכון השונות בתהליך, שכן "הבעייתיות" בפרוטוקול RADIUS הינה מורכבת מהרבה שלבים. בסוף גם עברנו קצת על הקונספט היחסית חדש של ZTNA.

על המחברת

אני דורין ראם, בת 21, משוחררת משירות של שלוש שנים ביחידת אופק חיל האוויר, בתפקיד מגנת סייבר כחוקרת אבטחה, בהתמקדות על רשת. כיום אני סטודנטית שנה א לתואר במדעי המחשב במכללה למנהל. מוזמנים לפנות אלי דרך הלינקדין שלי

מקורות

- https://docs.secureauth.com/2212/en/secureauth-security-advisory---radius-vulnerability-cve-2024-3596.html
- https://www.ietf.org/archive/id/draft-ietf-radext-deprecating-radius-08.html
- https://www.cloudradius.com/security-of-peap-mschapv2/
- https://www.tracenetsolutions.com/2024/09/27/radius-protocol-and-its-vulnerabilities/
- https://rcpmag.com/articles/2003/06/01/securing-wireless.aspx
- https://cyberinsider.com/radius-security-flaw-exposes-networks-to-forged-authentication-attacks/
- https://www.infoq.com/news/2024/07/radius-vulnerability/
- https://www.cisco.com/c/en/us/support/docs/security/identity-services-engine/222287-blast-radius-cve-2024-3596-protocol-sp.html
- https://www.cisco.com/c/en/us/support/docs/csa/cisco-sa-radius-spoofing-july-2024-87cCDwZ3.html
- https://www.cisco.com/c/en/us/support/docs/security/identity-services-engine/222287-blast-radius-cve-2024-3596-protocol-sp.html
- https://www.cve.news/cve-2024-3596/
- https://www.computing.co.uk/news/4334196/blast-radius-major-vulnerability-common-protocol
- https://www.computing.co.uk/news/4334196/blast-radius-major-vulnerability-common-protocol