מטלת סיום : קורס הגנת פרוטוקולי תקשורת

SSH vulnerability

מגישים:

יובל בר מעוז 314878877
yuvalbm3@gmail.com

0504550047

רז אלבז 207276775
raz4447@gmail.com

0526604785

תוכן עניינים

תיאור המשימה ורשימת כוונות

בחירת המשימה עבור מטלת החקר:

בחרנו במשימה 4.3: SSH vulnerability.

חלק תאורטי:

בחלק זה נתאר:

- פרוטוקול SSH-הגדרות, מונחים, שימושים, איך הוא פועל
- גרסאות פרוטוקול
- מפתחות SSH
- סקירת פגיעות ב SSH ברמה פנים ארגונית
- סקירת אבטחה בפרוטוקול SSH
- SSH Vulnerabilities ופתרונן
- Brute Force Attack
- בונוס משימה מספר 9- מחשב קוונטי

חלק מעשי:

כתיבת כלי תקיפה

בונוס ראשון שעשינו:

בחרנו במשימה:
9)מהן ההגנות ב SSH כנגד מחשבים קוונטים בעתיד ?

נושא שני: נושא מתקדם , אין מימוש אלא רק רקע תיאורטי , יכול לבוא כהשלמה לבונוס

בונוס שני שעשינו:

מעבדת SEED – מוגש בנפרד

פרוטוקול SSH

פרוטוקול SSH:

פרוטוקול SSH או בשמו המלא **Secure Shell** הוא פרוטוקול המשמש לתקשורת מחשבים ולביצוע פעולות על מחשב מרוחקת לאחר תהליך ה-login; תהליך ההזדהות של המשתמש במחשב. פרוטוקול SSH הוא שיטה לכניסה מאובטחת מרחוק ממחשב אחד למשנהו. הוא מספק מספר אפשרויות חלופיות לאימות חזק, והוא מגן על אבטחת התקשורת ושלמותה באמצעות הצפנה חזקה. זוהי חלופה מאובטחת לפרוטוקולי הכניסה הלא מוגנים (כגון telnet, rlogin) ושיטות העברת קבצים לא מאובטחות (כגון FTP).

פורט סטנדרטי:
פורט מספר 22

שימושים אופייניים בפרוטוקול SSH:
<u>הפרוטוקול משמש ברשתות ארגוניות עבור:</u>

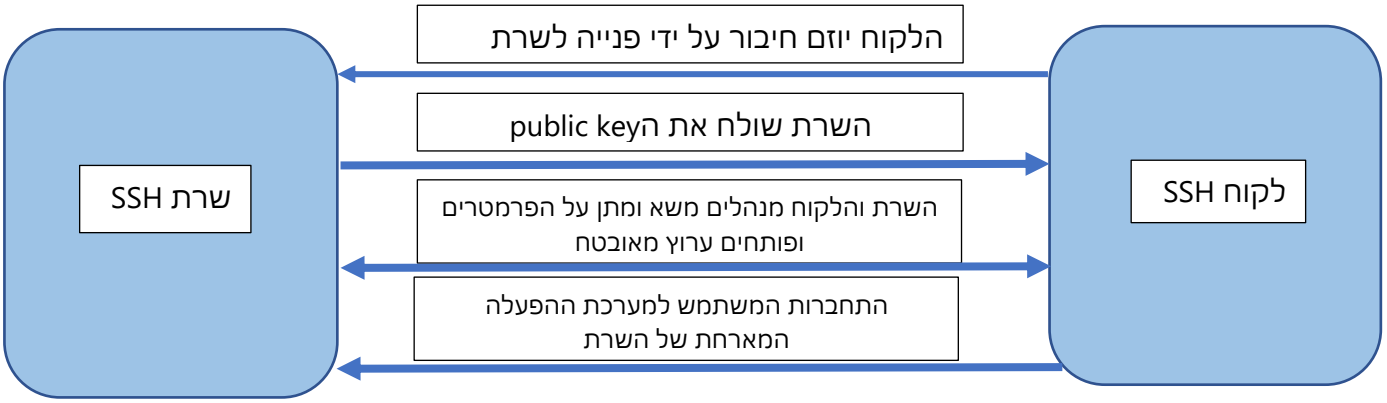
- מתן גישה מאובטחת למשתמשים ותהליכים אוטומטיים.
- העברות קבצים אינטראקטיביות ואוטומטיות.
- הוצאת פקודות מרחוק.
- ניהול תשתית רשת ורכיבי מערכת קריטיים אחרים.

כיצד פועל פרוטוקול SSH: (תיאור כללי פירוט בהמשך)

הפרוטוקול עובד במודל שרת-לקוח.
<u>בשלב הראשון</u>, הלקוח יוצר חיבור SSH לשרת SSH. לקוח SSH מניע את תהליך הגדרת החיבור ומשתמש בהצפנת מפתח ציבורי כדי לאמת את זהות שרת ה-SSH.

<u>בשלב השני</u>, לאחר שלב ההגדרה, פרוטוקול SSH משתמש בהצפנה סימטרית חזקה ובאלגוריתמי hashing כדי להבטיח את הפרטיות והשלמות של הנתונים המוחלפים בין הלקוח לשרת.

להלן תרשים המייצג את החיבור שתיארנו לעיל:



במהלך תהליך האימות, מפתחות SSH אלה מייצרים לעתים קרובות גישה ישירה, מיוחסת או גישה שורשית למגוון מערכות קריטיות, והופכים למעשה את נכסי ההצפנה הללו לאישורים מועדפים. למפתחות SSH ניתנת גישה זהה לסיסמאות, אך כאשר רוב האנשים חושבים על אבטחת האישורים המועדפים שלהם, הם שוכחים את מפתחות ה-SSH. כתוצאה מכך, מפתחות אלה יכולים ליפול בקלות לידיים הלא נכונות, ובמקום להגן על גישה לנכסים חשובים, מפתחות אלה יכולים להפוך ל"מפתחות שלד וירטואליים". כדי להחמיר את המצב, כאשר תוקף מקבל גישה למפתח SSH מיוחס אחד, הוא או הוא יכולים לגשת לכל מפתח SSH המאוחסן על המחשב הזה ולעבר את כל רשת החברה, ולעתים קרובות מקבלים גישה לכל נתוני החברה. רק חמישה עד 20 מפתחות SSH ייחודיים יכולים להעניק גישה לארגון שלם באמצעות אמון מפתח SSH טרנזיטיבי, המספק לתוקפים גישה מועדפת למערכות ולנתונים הרגישים ביותר של הארגון.

OpenSSH
יישום קוד פתוח של פרוטוקול SSH.
OpenSSH היא חבילת תוכנות מחשב המספקות גישה מרוחקת מאובטחת ומוצפנת על גבי רשת מחשבים על בסיס פרוטוקול SSH.

הלקוח המתקשר לשרת ה-OpenSSH מאמת את זהותו באמצעות שם משתמש וסיסמה שניתנו לו במעמד ההרשמה לשרת. התעבורה בין הלקוח לבין השרת מוצפנת בשיטת RSA על ידי מפתח ציבורי ומפתח פרטי. ישנם מספר תוכנות ליישום הלקוח של ה-openSSH המאפשרות חיבור SSH גם ממערכות הפעלה אחרות. כברירת מחדל משמש פורט 22 לתעבורה של פרוטוקול SSH.

OpenSSH כולל בתוכו את האפשרות ליצור מנהור* בין הלקוח והשרת, ולתעל תעבורה של חבילות TCP מהלקוח, היוצאות דרך אחד מהפורטים בשרת.
*מנהור- שימוש בפרוטוקול תקשורת המאפשר הכמסה של פרוטוקול אחר בתוכו, באופן שבו שדה המטען של פרוטוקול המעטפת מכיל הודעות המקודדות על פי פרוטוקול הליבה

אחד השימושים העיקריים בתוכנה זו הוא בגישה מרוחקת מאובטחת למעטפת (shell) של מערכות יוניקס, לינוקס ו-BSD, אך ישנן גם תוכנות המתבססות על אפשרות זו של מנהור והצפנת תעבורה לצורך העברת סוגים שונים של מידע בין מחשבים באופן מאובטח ומוצפן.

אלגוריתמי הצפנת SSH

SSH מצפין תעבורת רשת כדי למנוע האזנת סתר. קבוצת מילות המפתח הראשונה שאנו צריכים לזכור היא, טקסט רגיל וטקסט צופן. כל נתון בפורמט המקורי שלו נקרא טקסט רגיל. טקסט צופן הוא הגרסה המקושקשת של טקסט רגיל, שהיא חסרת משמעות עבור צד שלישי.

הצפנה ופענוח
הצפנה: היא תהליך המרת טקסט רגיל לטקסט צופן. במילים אחרות, ערבול טקסט רגיל.
פענוח: הוא רק תהליך הפוך, הממיר בחזרה את טקסט ההצפנה חסר המשמעות לפורמט הטקסט הרגיל המקורי שלו.

אלגוריתם הצפנה
מפתח הצפנה הוא בדרך כלל מחרוזת ארוכה. אלגוריתמי הצפנה משתמשים במפתח הצפנה כדי להצפין נתונים. אנחנו צריכים את אותו מפתח (להצפנה סימטרית), או מפתח הקשור מתמטית (להצפנה א-סימטרית) כדי לפענח את הנתונים המעורערים בחזרה לפורמט טקסט רגיל המקורי.

❤אלגוריתם ההצפנה לא מבטיח את המרת הנתונים המעורערים לגרסת הטקסט הפשוט שלו, חובה על אותו אדם לדעת מהו מפתח ההצפנה.

אלגוריתמי הצפנה סימטרית ואלגוריתמי הצפנה אסימטרית
הסט השלישי של מילות מפתח הקשורות לקריפטוגרפיה הם אלגוריתמי הצפנה סימטרית ואלגוריתמי הצפנה אסימטרית.
אלגוריתמים של הצפנה סימטרית-אלגוריתם הצפנה סימטרית משתמש באותו מפתח עבור הצפנה ופענוח.
לדוגמה: אם במהלך ההצפנה בחרת "RAZANDYUBALWORK" כמפתח ההצפנה הסימטרי להמרת טקסט רגיל לטקסט צופן. במהלך הפענוח, עליך לספק את אותו מפתח "RAZANDYUBALWORK " כדי להסתיר חזרה של טקסט צופן לטקסט רגיל.

דוגמאות לאלגוריתמים של הצפנה סימטרית הם:

- AES (Advanced Encryption Standard),
- Salsa20 / ChaCha20
- 3DES (Algorithm Triple Data Encryption)
- IDEA (International Data Encryption Algorithm)
- Blowfish.

אלגוריתמי הצפנה אסימטריים-
אלגוריתמי הצפנה אסימטריים משתמשים בקבוצה של מפתחות, המכונה בדרך כלל מפתח ציבורי ומפתח פרטי. מפתח ציבורי ומפתח פרטי הם מפתחות הקשורים מתמטית. ניתן לפענח את הנתונים המוצפנים במפתח אחד רק באמצעות המפתח הקשור המתמטית שלו. בהצפנת מפתח ציבורי, ניתן להפיץ מפתח ציבורי לכל מי שמבקש את המפתח הציבורי. ניתן לפענח את הנתונים המוצפנים במפתח ציבורי רק באמצעות המפתח הפרטי הקשור אליו, אשר נשמר מאובטח היטב במחשב שבבעלותו זוג המפתחות.

כמה דוגמאות לאלגוריתמים של הצפנה אסימטרית הם ,RSA (Rivest-Shamir-Adleman)

- ECC (Eliptic Curve Cryptography)
- DH (Diffie-Hellman)
- El Gamal
- ECDH (Eliptic Curve Diffie-Hellman)
- ECDSA (Elliptic Curve Digital Signature Algorithm)

<u>אלגוריתמי הצפנה ב-SSH</u>
אלגוריתמי הצפנה א-סימטרית עובדים לאט יותר מאלגוריתמי הצפנה סימטרית.
אלגוריתמי הצפנה א-סימטרית דורשים הרבה יותר כוח מחשוב מאלגוריתמי הצפנה סימטרית.

כעת הבעיה היא שלמרות שאלגוריתמי הצפנה סימטרית מהירים יותר, החלפת מפתח ההצפנה דרך הרשת עלולה לחשוף את המפתחות למשתמשים זדוניים. אז, SSH משתמש בהצפנה אסימטרית כדי ליצור מפתח סודי סימטרי משותף בין לקוח SSH לשרת SSH להצפנה ופענוח. מפתחות הצפנה סימטריים חדשים נוצרים, אם יש צורך בהעברת נתונים נוספים או שההפעלה נמשכת זמן רב, כדי להוסיף יותר אבטחה.

ל-SSH יש תמיכה באלגוריתמי הצפנה סימטריים ואסימטריים רבים ושונים. לקוח SSH ושרת SSH מנהלים משא ומתן ביניהם, אלגוריתמי ההצפנה לשימוש.

פגיעויות נפוצות SSH שחשוב להכיר:

1. בעיות מעקב מפתח SSH: ארגונים בדרך כלל צוברים מספר רב של מפתחות SSH מכיוון שמשתמשי קצה יכולים ליצור מפתחות SSH חדשים (אישורים) או אפילו לשכפל אותם ללא פיקוח, בניגוד לאישורים או סיסמאות.
ולכן, ברגע שמספר גדול של מפתחות SSH נבנה לאורך זמן, ארגון עלול, למשל, לאבד בקלות את האישורים הללו כאשר שרתי פיתוח מועברים לסביבות ייצור או כאשר עובדים עוזבים את החברה והמפתחות שלהם לא משתנים.
כתוצאה מכך, מפתחות SSH שנותרו יכולים לספק לתוקפים גישה מיוחסת לטווח ארוך למשאבים ארגוניים.
במידה וקורה מצב בו תוקף מקבל גישה למפתח שלעולם לא נשלל או מסובב, זאת יכולה להיות נקודת כניסה קבועה לרשת ולהתחזות למשתמש שמפתח ה-SSH שייך לו במקור.

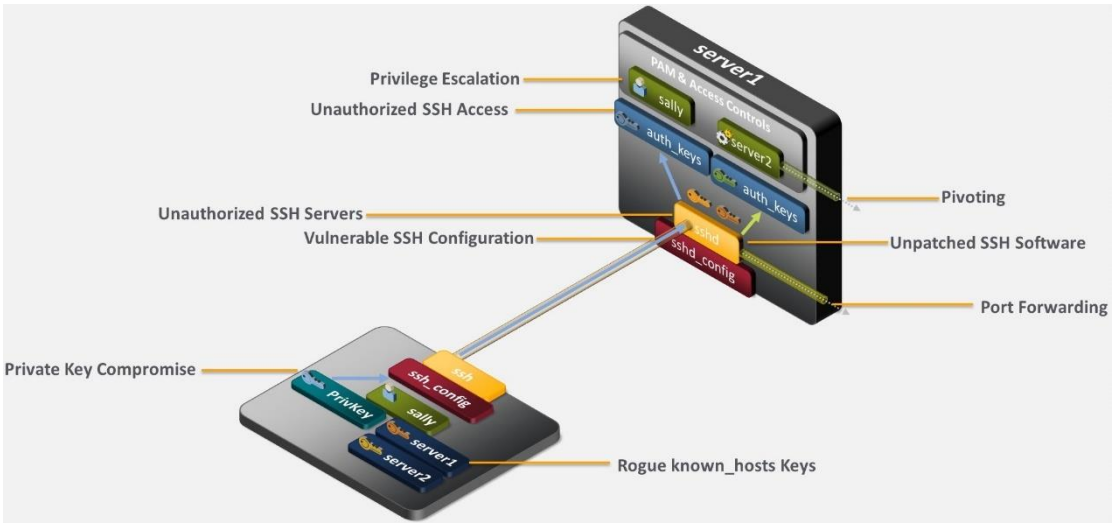
2.שיתוף מפתחות SSH:
מפתחות SSH משותפים או משוכפלים לרוב על פני קבוצה משותפת של עובדים או שרתים ורכיבי תשתית. שכפול מפתחות SSH מוביל לכך שקיימים 5-20 מפתחות ייחודיים אשר יכולים להעניק גישה לכל המכונות ברחבי הארגון.
בטווח הקצר: מקלה על העבודה של צוות IT
המטרה העיקרית של כל צוות IT היא לעזור לספק תוצאות עסקיות, כך שאנשי הצוות העסקי לא יצטרכו להפוך למומחי טכנולוגיה. צוותי IT מנהלים את הטכנולוגיה, כך שהצוות העסקי יכול להתמקד בפעילות העסקית.
בטווח הארוך: מקלה על עבודתם של תוקפים.
שכפול מפתחות SSH יוצרת מיפוי מפתח ציבורי פרטי סבוך ורבים לרבים, שמפחיתים משמעותית את האבטחה מכיוון שקשה לסובב ולבטל מפתח בודד מבלי לשבור קשרי מפתח SSH אחרים שלא ידועים להם שחולקים את אותה טביעת אצבע של מפתח. שיתוף מפתחות SSH הוא גם מסוכן מכיוון שהוא מפחית את יכולת הביקורת ואי-הדחה.

3.מפתחות SSH סטטיים:
מפתח SSH סטטי -שרת ה-SSH במארח המרוחק מקבל מפתח פרטי SSH סטטי ידוע בציבור לצורך אימות. תוקף מרוחק יכול להיכנס למארח זה באמצעות מפתח פרטי זה ידוע בציבור
חזרה של מיליון מפתחות SSH פלוס יהוו סיוט לוגיסטי. מנהלי IT ואנשי אבטחה רבים נמנעים משינוי או מיצירת מפתחות מחשש שרכיב קריטי או עובד עלול להישכח. כתוצאה מכך נגרמת עלייה במפתחות SSH סטטיים, ואותם אנשים פותחים את הדלת לתוקפים כדי לסכן מפתח ללא שינוי, להשתמש בו כדי לנוע לרוחב בארגון ולקבל גישה קבועה ובלתי מורשית לנתונים ולנכסים רגישים.

4.מפתחות SSH משובצים: אלה שאף אחד לא רוצה להתעסק איתם. מפתחות SSH מוטמעים לעתים קרובות בתוך יישומים או סקריפטים. מנהלי מערכת חוששים לעתים קרובות לשנות אותם מכיוון שהם אינם מבינים את הקוד שבו המפתחות מוטמעים או שנאסר עליהם מאוד לסובב אותם בגלל רמת התיאום הנדרשת כדי למנוע הפסקות מערכת. כתוצאה מכך, מפתחות SSH סטטיים המוטמעים ביישומים, בקוד ובסקריפטים יכולים להוביל לדלתות אחוריות מתמשכות לתוקפים.

מפתחות SSH יכולים להוות הזדמנות אדירה לתוקפים לקבל גישה מועדפת לרשתות, להישאר מחוברים, להתחזות למשתמשים לגיטימיים, להסתיר את פעילותם בהצפנה ולנוע בחופשיות.

חסרונות וסכנות בשימוש בפרוטוקול SSH:



<u>שרתי SSH לא מאושרים</u>
אם יש לארגון מסוים משתמשים ומנהלי מערכת המאפשרים גישת שרת (sshd) SSH במערכות שבהן היא אינה נדרשת, הם מרחיבים את משטח ההתקפה מכיוון שלתוקפים תהיה אפשרות גדולה יותר לקבל גישה מרחוק למערכות אלו.

<u>תוכנת SSH לא מתוקנת</u>
עבור המערכות שבהן השימוש ב-SSH מוצדק, אם תוכנת שרת SSH ותוכנת לקוח לא מתעדכנת בתיקונים ועדכונים, היא עלולה לחשוף את המערכות והנתונים שהיא נועדה להגן עליהן ולהפוך אותן לפגיעות לפגיעה.

<u>תצורת SSH פגיעה</u>
רוב יישומי שרת ה-SSH והלקוחות כוללים מספר לא מבוטל של פרמטרי תצורה המשפיעים על תפעול ואבטחה, כולל אפשרויות לאימות, גישת שורש, העברת יציאות, מיקומי קבצים וכו'. במהלך השנים, רוב מפתחי הטמעת SSH בחרו תצורות ברירת מחדל שהן מאובטחות יותר. עם זאת, ישנן כמה ברירות מחדל, כגון העברת יציאות ומיקום קבצי מפתח מורשים, שאינן אופטימליות. בנוסף, אם המשתמשים והמנהלים שלך משנים באופן שרירותי את התצורות הללו מבלי להתחשב בהשלכות האבטחה, הם יכולים לפתוח את המערכות הללו להתקפות רחבות יותר.

<u>העברת יציאות SSH</u>
עוד מהימים שבהם ההצפנה לא הייתה זמינה עבור כל הפרוטוקולים, SSH כולל את היכולת להעביר תעבורה שנשלחה ליציאה מקומית בלקוח SSH. התעבורה מועברת דרך סשן ה-SSH המוצפן לשרת ה-SSH או אפילו מעבר לכך. האתגר הוא שזה מספק את היכולת לתקשורת לא מאושרת לחצות חומות אש. אם המשתמש של לקוח SSH שקיבל גישת SSH לשרת בצד השני של חומת האש רשאי לאפשר העברת יציאות מקומית, הוא פותח את האפשרות שתוקף יכול לקבל גישה למערכות והתקנים שאחרת לא היו נגיש. על ידי ניצול העברת פורטים, תוקף עלול לעקוף חומות אש שהוגדרו להגבלת הגישה לרשת השרת. בנוסף, התוקפים נמנעים מזיהוי מכיוון שהם פועלים בחיבור SSH מוצפן.

פשרה על מפתח פרטי

כאשר אתה מגדיר SSH לאימות מפתח ציבורי, מפתחות פרטיים מאפשרים גישה לחשבונות. אם מפתח פרטי נפגע, תוקף יכול לבצע אימות לחשבונות שבהם המפתח הפרטי הוא מהימן. להלן כמה מהסיכונים הנשקפים למפתחות פרטיים של SSH:

- משתמשים חסרי זהירות: כאשר משתמשים מורשים להשתמש באימות מפתח ציבורי SSH, הם יכולים להיות רשלניים בטיפול במפתחות הפרטיים שלהם, בין אם למקם אותם במיקומים לא מאובטחים, להעתיק אותם למספר מחשבים, ולא להגן עליהם בסיסמאות חזקות.
- תחלופה של מנהל מערכת: כאשר נעשה שימוש באימות מפתח ציבורי לתהליכים אוטומטיים, מנהל אחד או יותר של התהליך יהיה אחראי על ניהול המפתח הפרטי של התהליך. מנהלי מערכת יכולים ליצור עותקים של המפתחות הפרטיים הללו, ואם הם יוקצו מחדש או נסתיימו, יכולים להשתמש במפתח/ים כדי לאמת את שרתי היעד.
- מפתחות חלשים: מכיוון שמפתחות SSH רבים לא שונו במשך שנים, מפתחות באורך קטן יותר (לדוגמה, מפתחות של 512 או 768 סיביות) עדיין נמצאים בשימוש, מה שמאפשר לתוקף מתוחכם לגזור את הערך של המפתח הפרטי. בנוסף, היו באגים בספריות קריפטוגרפיות שהביאו ליצירת מפתחות חלשים וניתנים לשבירה בקלות.

גישת SSH לא מורשית

מכיוון ש-SSH מספקת גישה מרחוק למערכות, חיוני שהגישה תהיה במעקב ובקרה. מכיוון שלארגונים רבים אין פיקוח ובקרה מרכזיים על SSH, הסיכון לגישה לא מורשית הולך וגדל. להלן כמה מהסיכונים הללו:

- יחסי אמון ללא מעקב: בדיוק כמו בכל שיטת גישה, אי שמירה על מלאי של היכן מותקנים מפתחות SSH ויחסי האמון שהם יוצרים בין מערכות וחשבונות היא מתכון לגישה לא מורשית. כשמנהלי מערכת באים והולכים עם הזמן, ארגונים רבים צברו מספר רב של מפתחות SSH אך אין להם גישה לגישה שהם מספקים.
- עובדים שהופסקו: אם משתמשי SSH - בין אם עובדים ובין אם קבלנים חיצוניים - משנים תפקידים או מופסקים והגישה שלהם לשרתי SSH לא מתעדכנת או נסגרה כהלכה, לאנשים אלה יכולה להיות גישה מתמשכת (עם זאת לא מורשית) למערכות קריטיות למשימה.
- מפתחות דלת אחורית: כברירת מחדל, רוב יישומי SSH (למשל, OpenSSH) מאפשרים למשתמשים להגדיר קבצי מפתח מורשים משלהם (הצבת מפתח ציבורי בחשבון כדי שיוכלו לגשת אליו באמצעות מפתח פרטי). אם ארגונים לא מחזיקים מלאי מעודכן של מפתחות מורשים ובודקים אותו באופן קבוע, משתמשים או אפילו תוקפים עשויים למקם מפתחות מורשים במקומות לא צפויים לגישה עתידית.

הסלמה של הרשאות

SSH משולב בדרך כלל עם רכיבים אחרים כדי לאפשר גישה. קשה מספיק לתזמר מרכזית את התצורה המאובטחת של כל הרכיבים הללו כדי למנוע מהתוקף להסלים הרשאות בהצלחה במהלך התקפה. זה אפילו יותר מאתגר כשיש לך מספר מנהלים בודדים שכל אחד מהם מקבל החלטות לגבי הטמעת SSH ללא כל פיקוח או ביקורת מרכזיים. ללא פיקוח זה, תעמוד בפני פוטנציאל גדול יותר להסלמה של הרשאות, במיוחד מכיוון

שלתוקפים הניגשים מרחוק למערכות באמצעות SSH יש הפעלה מוצפנת שבתוכה להסתיר את פעולותיהם.

מפתחות מארח ידועים של נוכלים
אם משתמש או מנהל שיוצרים תחילה חיבור מלקוח SSH לשרת SSH אינם בודקים את האותנטיות של המפתח הציבורי עבור אותו שרת, הם עשויים לקבל מפתח ציבורי של תוקף ולאפשר התקפה.

יתרונות בשימוש בפרוטוקול SSH:

SSH מאפשר הצפנת נתונים כך שאותם תוקפים זדוניים לא יוכלו לגשת לפרטי המשתמש ולסיסמאות שלך. SSH מאפשר גם לבצע מנהור של פרוטוקולים אחרים כגון **FTP**- (File Transfer Protocol) הוא פרוטוקול תקשורת מבוסס TCP להעברת קבצים בין מחשבים. באמצעות פרוטוקול זה, תוכנת לקוח FTP מתקשרת עם תוכנת שרת FTP, לשם לקיחת קובץ מהשרת או הוספת קובץ אליו. שימושים אופייניים לפרוטוקול: הורדת קובצי מולטימדיה למיניהם מאתר המאחסן קבצים כאלה.

להלן רשימה של דברים ספציפיים שמהם SSH מגן:

<u>ניתוב מקור IP</u>
בעוד שבדרך כלל משתמשים בניתוב מקור למטרות טובות כמו שינוי הנתיב של אות רשת אם הוא נכשל במקור, הוא יכול לשמש גם משתמשים זדוניים כדי לגרום למכונה לחשוב שהיא מדברת עם מכשיר אחר.

<u>זיוף DNS</u>
זהו סוג של התקפת פריצה שבה נתונים מוכנסים למסד הנתונים של שרת השמות של מערכת שמות דומיין. זה גורם לשרת השמות להחזיר כתובת IP שגויה כדי שיוכל להפנות תנועה למחשב אחר. לרוב זה המחשב של התוקף. משם הם יכולים לקבל מידע רגיש.

<u>מניפולציה של נתונים בדברים כמו נתבים לאורך הרשת</u>
זה די מובן מאליו, התוקף משיג או משנה נתונים אצל מתווכים לאורך תוואי הרשת. זה מבוצע לרוב בנתבים שבהם נתונים נכנסים למעין שער או מחסום בדרך ליעדם.

<u>ציתות או רחרוח של הנתונים המועברים</u>
אם משתמשים בחיבור לא מאובטח, תוקף יכול לצפות בנתונים שעוברים, לאסוף כל מיני מידע רגיש או פרטי לשימושים זדוניים משלו.

<u>זיוף כתובות IP</u>
זיוף IP הוא הזמן שבו משתמש זדוני יוצר מנות עם כתובת IP מזויפת מקור. כך הוא שומר על זהות מחשב המקור ומיקומו בסוד ונראה שהוא מחשב אחר שהמקלט סומך עליו.

<u>גיבויים והגירות יעילים</u>
מכיוון שכל הגיבויים וההגירה יכולים להתרחש ישירות בשרת האינטרנט, הדברים מתנהלים במהירות הבזק.

<u>תקופת תוקף וביטול</u>
יתרון מרכזי של תעודות SSH הוא תקופת התוקף. הם תקפים לתקופה מסוימת בלבד ולאחר מכן לא ניתן יהיה לסמוך עליהם יותר. תקופת התוקף יכולה להיות ימים, שעות ואפילו דקות.
זה אפילו טוב יותר כאשר משתמשים בתעודות קצרות מועד. מנהלי מערכת לא יצטרכו עוד לוודא שרשימות הביטול מאוכלסות באישורים שבוטלו. אם עובד יאבד גישה (למשל, אם הוא עוזב את החברה), האישור הקיים שלו יפוג, והוא לא יוכל לקבל אישור חדש. ביטול פסיבי זה

מהווה יתרון במקרה של מפתח פרטי שנפגע. אם יש מחשב נייד שאבד או נגנב, תעודת SSH קצרת מועד אינה שווה ערך עבור גישה לתשתית פנימית על ידי צד שלישי לא מורשה.

כניסה ויציאה למשתמשים

ארגונים רבים תיארו תהליכים של כניסה ויציאה למשתמשים הכוללים גישת SSH. זה לא מפתיע שתקנות האבטחה סביב תהליכים אלה לא תמיד מקובלות בקפדנות. במהלך השנים, כאשר מנהלי מערכת עוזבים ומנהלים חדשים מגיעים להצטרף, מפתחות ישנים נשארים מאחור. לא לדעת אילו מפתחות ישנים נשארו מאחור היא בעיה גדולה, מכיוון שלא סביר שמנהלי מערכת חדשים יוכלו ליצור מחדש ולהחליף את כל המפתחות הקיימים שנמצאים בשימוש.

בעת שימוש בתעודות SSH, תהליך ההטמעה מפושט כדי להנפיק אישור חדש עבור כל מנהל מערכת חדש. מכיוון ששרתים סומכים על רשות אישורים, במקום מפתחות ציבוריים בודדים, אין צורך להגדיר כל שרת.

כאשר יוצאים ממשתמשים שמשתמשים באישורים קצרי מועד, אתה יכול פשוט לאפשר להם לפוג מבלי לחדש אותם, וניתן לבטל אישורים אם הם תקפים לפרקי זמן ארוכים.

אמון בשרתים מרוחקים

כאשר לקוח מתחבר לשרת בפעם הראשונה, המשתמש מתבקש לאמת את המפתח הציבורי של השרת. לאחר קבלתו, הלקוח לא יבקש שוב את המשתמש אלא אם יש מפתח ציבורי חדש. תהליך זה נקרא Trust on First Use (TOFU).

בעת שימוש במפתחות SSH מסורתיים, עליך לשים את המפתחות הציבוריים של כל השרתים שלך בכל מכשירי הלקוח. זה לא עניין גדול אם ארגונים משתמשים בשרתים של bastion (המכונה קפיצה). אחרת, זה יכול להיות אתגר.

אמון ברשות אישורים על ידי הוספת המפתח הציבורי שלה הוא תהליך פשוט מאוד. זה יאפשר לארגונים להוסיף שרתים חדשים או להחליף אותם ללא מנהלי מערכת כדי לראות את הודעות השגיאה "מפתח המארח השתנה".

תעודות מארח מאפשרות למארחים מרובים לשתף עיקרון מבלי צורך לשתף את אותו מפתח פרטי, וזה שימושי להפליא בעת שימוש בחיבורי SSH מאוזני עומס.

גרסאות פרוטוקול SSH

יש בעיקר שתי גרסאות של פרוטוקול SSH. הגרסה הראשונית הייתה SSH-1, אשר שוחררה ביולי 1995. בשנת 2006, (Internet Engineering Task Force) IETF פרסם RFCs עבור גרסה מתוקנת של פרוטוקול SSH SSH-2 כסטנדרט. שתי הגרסאות של SSH, SSH-1 ו-SSH-2 אינן תואמות.

SSH-1 (גרסה ראשונית של SSH)

הגרסה הראשונה של SSH הייתה SSH-1. SSH-1 הייתה הגרסה הראשונית של SSH שהומצאה על ידי Tatu Ylonen, באוניברסיטת הלסינקי לטכנולוגיה, פינלנד. הסיבה מאחורי המצאת SSH-1 הייתה התקפת sniff סיסמאות באוניברסיטה על פרוטוקול פחות מאובטח. Tatu Ylonen הוציא את הגרסה הראשונה של SSH ביולי 1995, כפרוטוקול חינמי. Tatu Ylonen הוציאה גם מוצר תוכנה, המבוסס על פרוטוקול זה. טאטו ילונן ראתה מיד עלייה במספר המשתמשים בפרוטוקול SSH. עד סוף 1995, היו בסך הכל 20,000 משתמשים עבור SSH. יש כמה פרצות אבטחה חמורות עם SSH-1.

SSH-2 (גרסה סטנדרטית של SSH)

SSH2 הוצג בשנת 2006 כתקן על ידי IETF. ל-SSH-2 שיפורים משמעותיים רבים ביחס ל-SSH-2. SSH1 מונע פרצות אבטחה רבות של SSH2. SSH-1 הוא הרבה יותר בטוח ויעיל מ-SSH-2. SSH-1 תומך ב-SFTP, גרסה מאובטחת של FTP. הנקודה העיקרית שיש לציין היא ש-SSH-1 ו-SSH-2 הם פרוטוקולים שונים לחלוטין. SSH-2 תוכנן לחלוטין כחדש, מההתחלה. SSH-2 היא הגרסה הנפוצה ביותר של פרוטוקול SSH בימינו.

SSH-1.99 (גרסת תאימות לאחור של SSH)

SSH-1.99 פורסם גם כתקן בשנת 2006 כ-RFC 4253. כפי שצוין קודם לכן, SSH-1 ו-SSH-2 אינם תואמים זה לזה. מטרת SSH-1.99 היא לספק תאימות לאחור עבור SSH-2 עם SSH-1.

מפתחות SSH

מפתחות SSH, או בשמן המלא **Secure Shell** הם אישורי אימות בפרוטוקול SSH.
מבחינה טכנית, הם מפתחות קריפטוגרפים (מלשון **קריפטוגרפיה** .תורת ההצפנה - ענף לשיטות באבטחת מידע) האחראיים להצפנה.
מבחנה מעשית, הדרך להשתמש בהן היא בצורה שדומה לסיסמאות.

SSH מספק הגנה חזקה על הצפנה ושלמות. לאחר שנוצר חיבור בין לקוח SSH לשרת, הנתונים המועברים מוצפנים בהתאם לפרמטרים שנקבעו בהגדרה. במהלך המשא ומתן, הלקוח והשרת מסכימים על אלגוריתם ההצפנה הסימטרי שישמש וייצור את מפתח ההצפנה שישמש. התעבורה בין הצדדים המתקשרים מוגנת באמצעות אלגוריתמי הצפנה חזקים בתקן התעשייה, ופרוטוקול SSH כולל גם מנגנון המבטיח את שלמות הנתונים המועברים באמצעות אלגוריתמי hash סטנדרטיים.

אימות חזק עם מפתחות SSH:
ישנן מספר אפשרויות שניתן להשתמש בהן עבור אימות משתמש. הנפוצים שבהם הם: סיסמאות ואימות מפתח ציבורי.

אנחנו נדון באפשרות האימות החזקה באמצעות מפתח ציבורי ועליו נפרט:
שימושים עיקריים:

1. השימוש העיקרי-אוטומציה; היא שימוש באביזרים מכניים או אלקטרוניים, על-מנת לבצע סדרת פעולות, ברצף מתוכנן, ללא מגע יד אדם. האוטומציה משחררת את האדם, בין היתר, מביצוע מטלות שגרתיות וחזרתיות, והן מוחלפות באמצעים טכנולוגיים עצמאיים.
2. שימוש נוסף בשיטת אימות זו נעשית על ידי מנהלי מערכת לכניסה יחידה.
3. העברות אוטומטיות של קבצי מעטפת מאובטחות משמשות לשילוב חלק של יישומים.
4. העברת שערים, המכונה לעיתים "תיעול", היא פעולת ההעברה של פתחה ממכונה אחת לאחרת. דרך אחת לניצול שיטה זו היא לצורך גישה למכונה ברשת פרטית דרך מחשב מחוץ לרשת הזו. הדבר מאפשר למחשבים מרוחקים, אשר מחוברים לרשת האינטרנט, למשל, להתחבר למכונה מסוימת המחוברת לרשת פרטית, שאינה נגישה בדרכים רגילות.
5. לניהול מערכות ותצורה אוטומטיות
6. כניסה מרחוק
7. ביצוע פקודות מרחוק
8. כניסות אוטומטיות
9. גיבוי
10. העתק ושיקוף קבצים
11. העברת קבצים מרחוק
12. VPN מוצפן מלא עבור שרתי OpenSSH ולקוחות התומכים בתכונה זו
13. ניטור וניהול מרחוק של שרתים באמצעות הכלים המפורטים לעיל

בתוך ארגון, מפתחות SSH משמשים בעיקר להענקת גישה מאובטחת למערכות מרוחקות. הרעיון הוא לקבל זוג מפתחות קריפטוגרפים - מפתח ציבורי (public key) ומפתח פרטי (private key)- ולהגדיר את המפתח הציבורי בשרת כדי לאשר גישה ולהעניק לכל מי שיש לו עותק של המפתח הפרטי גישה לשרת. המפתחות המשמשים לאימות נקראים **מפתחות SSH**. אימות מפתח ציבורי משמש גם עם כרטיסים חכמים, כגון כרטיסי CAC ו- PIV המשמשים את ממשלת ארה"ב.

מפתחות ציבוריים או מפתחות מורשים אחראים להענקת גישת כניסה למשתמשים שניגשים למערכת המרוחקת. אפשר לדמיין מפתחות מורשים כמנעולים, המעניקים גישה למי שמחזיק במפתח הנכון (במקרה זה, המפתח הפרטי המתאים). מפתחות מורשים מוגדרים בנפרד עבור כל חשבון משתמש, והם נמצאים בדרך כלל בקובץ ~/.ssh/authorized_keys בספריית הבית של המשתמש.

מפתחות פרטיים או מפתחות זהות מאפשרים למשתמשים לאמת את עצמם בשרת SSH. מפתח פרטי הוא אנלוגי למפתח אמיתי שעם התאמה יכול לפתוח מנעול אחד או יותר. בניגוד למפתחות ציבוריים, מפתחות פרטיים צריכים להיות מאובטחים במקום מאובטח או שהם עלולים להגיע לידיים הלא נכונות, וכתוצאה מכך שימוש לרעה בהרשאות.

תהליך אימות מפתח ssh:

אימות מפתח ציבורי מבוסס SSH מושג על ידי יצירת זוג מפתחות נפרדים (מפתח ציבורי ופרטי) על מנת ליצור קשר עם מערכות מרוחקות. המשתמש המאמת את עצמו במחשב המרוחק צריך להחזיק את המפתח הפרטי, בעוד שהמפתח הציבורי צריך להיות ממוקם במערכת/מערכות היעד שאליהן המשתמש רוצה להתחבר.

- צור מפתחות אימות SSH: היכנסו לשרת שממנו יש ליצור את החיבור המרוחק. צרו זוג מפתחות באמצעות כלי ליצירת מפתחות SSH. הנכם יכולים להוסיף ביטוי סיסמה אופציונלי בזמן יצירת צמד המפתחות כדי לספק שכבת אבטחה נוספת.
- העתקת המפתחות הציבוריים למערכות מרוחקות: לאחר יצירת צמד המפתחות, העבירו את המפתחות הציבוריים למערכות יעד מרוחקות. ודא שקובצי המפתח הציבורי ממוקמים תחת ספריית ~/.ssh/authorized_keys בשרתים המרוחקים הדרושים. כאן, תצטרכו לספק את ביטוי הסיסמה שנוצר בשלב הקודם.
- הפעילו הפעלות SSH מרוחקות: לאחר שפרסתם את המפתחות הציבוריים לשרתי היעד, תוכלו לפתוח חיבורי SSH עם אותם שרתים מהמערכת שלך.

איך תהליך אימות מפתח SSH מתרחש ברקע:

1. הלקוח מתחיל בשליחת מזהה לזוג המפתחות שהוא רוצה לאמת בשרת המרוחק.
2. השרת בודק אם ישנם מפתחות ציבוריים בעלי אותו מפתח בחשבון שאליו הלקוח מנסה להיכנס.
3. אם נמצא המפתח הציבורי התואם, השרת יוצר מספר אקראי, מצפין אותו במפתח הציבורי ושולח אותו ללקוח.
4. הלקוח מפענח את ההודעה עם המפתח הפרטי ובעזרת מפתחות הפעלה הוא מחשב את ערך הגיבוב MD5 של ההודעה.
5. לאחר מכן הלקוח מצפין את ערך ה-hash ושולח אותו לשרת.
6. בינתיים, השרת גם מחשב את ערך הגיבוב MD5 של ההודעה שנשלחה ללקוח (בעזרת מפתחות הפעלה). אם שני הערכים הללו תואמים, זה מוכיח שללקוח יש את המפתח הפרטי המתאים, והלקוח מאומת בשרת.

הערה: לפני תחילת תהליך האימות של מפתח SSH, ודא שגם בלקוח וגם בשרת מותקנת גרסה עובדת של SSH. יתרון גדול אחד של אימות מבוסס מפתח הוא שכאשר הוא מיושם בצורה נכונה, הוא מפשט מאוד את תהליך האימות ומגביר את האבטחה שלו פי כמה.

סיבות לאבטחת מפתחות ה-SSH:

בממוצע, ארגון גדול מכיל בקלות כמיליון מפתחות SSH. ולמרות המודעות שנוצרה סביב ניהול לא נכון של מפתח SSH וסיכונים, רוב הארגונים נוקטים בגישה מבוזרת ליצירת מפתח ושימוש בהם. לאורך זמן, זה מביא להתרבות מפתחות עם הרבה פחות נראות לגבי מידת הגישה שכל מפתח מספק. ניהול כושל של מפתח SSH יכול להביא צרות לארגונים בדרכים רבות, כולל הבאות:

התקפה חיצונית: כאשר תוקף מקבל גישה למפתח SSH יתום, הוא מסוגל לבסס דריסת רגל ולנוע בקלות בתוך הרשת, מכיוון שרשתות מבוססות מפתח שזורות היטב. לאחר מכן, התוקף יכול להעלות את ההרשאות שלו לגישה לשורש, לשאוב נתונים רגישים ממערכות קריטיות למשימה וליצור דלתות אחוריות לגישה קבועה.

מתקפת פנים: ניהול לקוי של מפתחות SSH הוא גם תורם מרכזי להתקפות פנימיות. עובדים ממורמרים או ספקים או קבלנים זדוניים של צד שלישי יכולים לקבל גישה למפתחות המאמתים מערכות מיוחסות ובסופו של דבר לבזוז נתונים רגישים.

דבר מסובך אחד בהתקפות האלה הוא שמכיוון שמפתחות SSH מסתובבים לעתים רחוקות, הם ימשיכו להישאר בידיים הלא נכונות עד שהם יתגלו איכשהו והגישה שלהם תופסק.

סיכוני אי ציות: ניהול מפתחות SSH חיוני גם עבור ארגונים כדי להציג ציות לתקנות מחייבות שונות בתעשייה כגון SOX, FISMA, PCI ו-HIPAA. אי שילוב מערכת ניהול מפתחות SSH נכונה עלול לגרום לחברות להיתקל בדרישות הציות ולעלות להן בקנסות אדירים.

ניהול מפתחות SSH:

תחילת העבודה:

ניהול מפתחות SSH הוא תהליך של אבטחה ואוטומציה של מחזור החיים של מפתחות SSH המופצים על פני ארגון - החל מיצירתם ועד לפריסתם לנקודות קצה נחוצות, השקת הפעלות מרוחקות, ניטור מיפויי מפתח-משתמשים, ביצוע סיבוב מפתחות ומחיקה מעת לעת של המפתחות שאינם בשימוש. או מפתחות לא רצויים. מפתחות SSH שייכים לקטגוריה של נכסים דיגיטליים מכריעים המנוהלים כל הזמן בחוסר. יישום תהליך ניהול מוגדר היטב מסייע לארגונים להשיג נראות מלאה על סביבת ה-SSH שלהם ולמנוע שימוש לרעה בהרשאות הנובעות מגישה לא מורשית למפתחות SSH.

1. הצעד הראשון בניהול מפתחות ה-SSH שלך הוא לגלות את המפתחות הקיימים ברשת שלך ולאחד אותם במאגר מרכזי.

2. מפה את יחסי האמון

לאחר שאספת את כל מפתחות ה-SSH שלך במקום אחד, עליך להתחקות אחר קשרי האמון הקיימים כדי לקבל תמונה ברורה של מידת הגישה שכל מפתח מעניק. לאחר מכן, עליך להגדיר את מיפוי מפתח משתמש, שיעזור לך לזהות את מספר המשתמשים ברשת שלך שיש להם גישת שורש לחשבונות מועדפים.

3. צור ופרוס צמדי מפתחות SSH טריים

כעת, לאחר שיש לך ראות מלאה על סביבת ה-SSH שלך, סרוק והסר מפתחות SSH שאינם בשימוש עוד. הגישה המומלצת כאן היא להתחיל מחדש על ידי מחיקת כל מפתחות ה-SSH המשויכים לחשבונות המשתמש השונים ברשת שלך, ולהחליף אותם בצמדי מפתחות טריים שנוצרו.

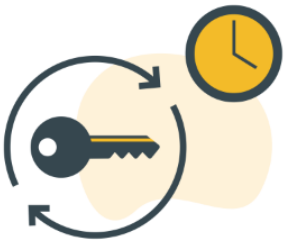
4. ייעל את היצירה והפריסה של מפתח SSH

ליצירת מפתחות, מומלץ מאוד לייעל את התהליך על ידי ניהול מסגרת מרכזית המאפשרת רק למשתמשים בעלי הרשאות ספציפיות ליצור ולפרוס מפתחות למערכות בתוך הרשת שלך. בדרך זו, יש לך יד על העליונה ביחסי האמון בתוך הארגון שלך, ואתה יכול לשמור על התפשטות מפתחות SSH.

5. הטלת בקרות גישה מפורטות

לאחר פריסת צמדי מפתחות חדשים למערכות יעד, חשוב להגדיר הרשאות עבור כל מפתח על סמך תפקידי המשתמש, כלומר הגבלות על המארח שממנו ניתן להשתמש במפתח, ואילו פקודות המפתח יכול לבצע.

6. לאכוף סיבוב מפתח SSH תקופתי
סיבוב מפתח SSH הוא הנוהג של לזרוק את שיוך מפתח-משתמש SSH הקיים ופריסה של צמדי מפתחות חדשים מעת לעת כדי להילחם באירוע המצער של פגיעה במפתחות SSH. יש לסובב מעת לעת גם את המפתחות המורשים (המוצבים במערכות היעד) וגם את מפתחות הזהות כדי למנוע שימוש לרעה בהרשאות.



7. בדוק את כל פעילויות המשתמש והפק דוחות משומרים
הגדר מנגנון ביקורת חסין חבלה כדי לעקוב אחר כל פעילויות המשתמש הכוללות מפתחות SSH ולהפיק דוחות מסווגים. זה ישפר את הרגישות והקריאות של הנתונים שנאספו ויעזור למנהלי מערכת לקבל החלטות עסקיות מושכלות.

סקירת פגיעות ב SSH ברמה פנים ארגונית

לאחר שנחשפנו לפרוטוקול SSH, לשימושים שלו, לגרסאות שלו, ליתרונות ולחסרונות שלו נרצה להראות כיצד ניצול חולשותיו של הפרוטוקול עלולים לשמש לתקיפה.
ראשית, לפרוטוקול SSH כמו לפרוטוקולים אחרים ישנם חסרונות שהצליחו להתגבר עליהם ויש חסרונות שטרם הצליחו. מטרת תוקף היא לנצל את האחרונים בכדי לממש את התקיפה ולצלוח את מטרותיו.

אם האקרים מקבלים גישה למארח המריץ SSH ברשת הפנימית של ארגון, הם יכולים להשתמש ב-SSH על אותו מארח כצינור מאובטח לניצול שירותי רשת פרטיים, לכן חיוני להבין כיצד ניתן להשתמש בגישה ל-SSH.

<u>תקיפה על ידי סריקת מפתחות SSH</u>
ממאמר שנקרא "האקרים סורקים כעת מפתחות SSH לניצול" נחשפנו לניצול של הפרוטוקול SSH בכדי לתקוף ארגון.

במאמר סופר על הדרך שבה האקרים מצליחים להיכנס לשרתים. היות ופרוטוקול Secure Shell ומפתחות SSH נמצאים בכל מקום במרכזי נתונים ושרתים בכל פינה בעולם, סופר בפירוט על ההתמודדות של התוקפים עם מפתחות SSH במקום השימוש בסיסמאות קלות לפיצוח שהיו בעבר לאותם שרתים.
בעבודתם עם Marist college הם למדו על הדרך שבה האקרים מנצלים ניהול לקוי של מפתחות SSH וזה באמצעות למידה על דרך הפעולה של אותם תוקפים עם המפתחות הללו.
באמצעות פעילות שדווחה על ידי שרתי אינטרנט הוכיחה שתוקפים מנצלים מפתחות SSH כדי לקבל גישה לנתוני החברה. תוקפים יכולים לפרוץ את ההיקף במספר דרכים, כפי שהם עשו, אבל ברגע שהם נכנסים, הם גונבים מפתחות SSH כדי לקדם את ההתקפה. רוב העסקים יודעים זאת ונקטו בצעדים כדי לטפל בבעיה זו, אך חלק מהחברות עדיין איטיות לאמץ את השיטות הטובות ביותר של SSH ותאימות.

תוכנות כופר, תוכנות זדוניות והונאות fishing עולות לכותרות, אבל ההתקפות הללו הן רק ההתחלה. הבעיה הגדולה יותר היא גישת SSH לא מבוקרת. פעילות האקרים אחרונה מראה שתוקפים סורקים באופן פעיל שרתי אינטרנט למפתחות SSH לניצול.

ייעוץ Wordfence האיר זרקור על סריקת מפתח המונית:

Wordfence, חברת אבטחה, זיהתה מתקפה חדשה המתמקדת בסריקה המונית של אתרי אינטרנט עבור מפתחות SSH פרטיים. Wordfence פרסמה ייעוץ כדי להבטיח שקהילת האינטרנט הרחבה יותר מודעת לפעילות החדשה הזו ולאיום הברור שהיא מייצגת.

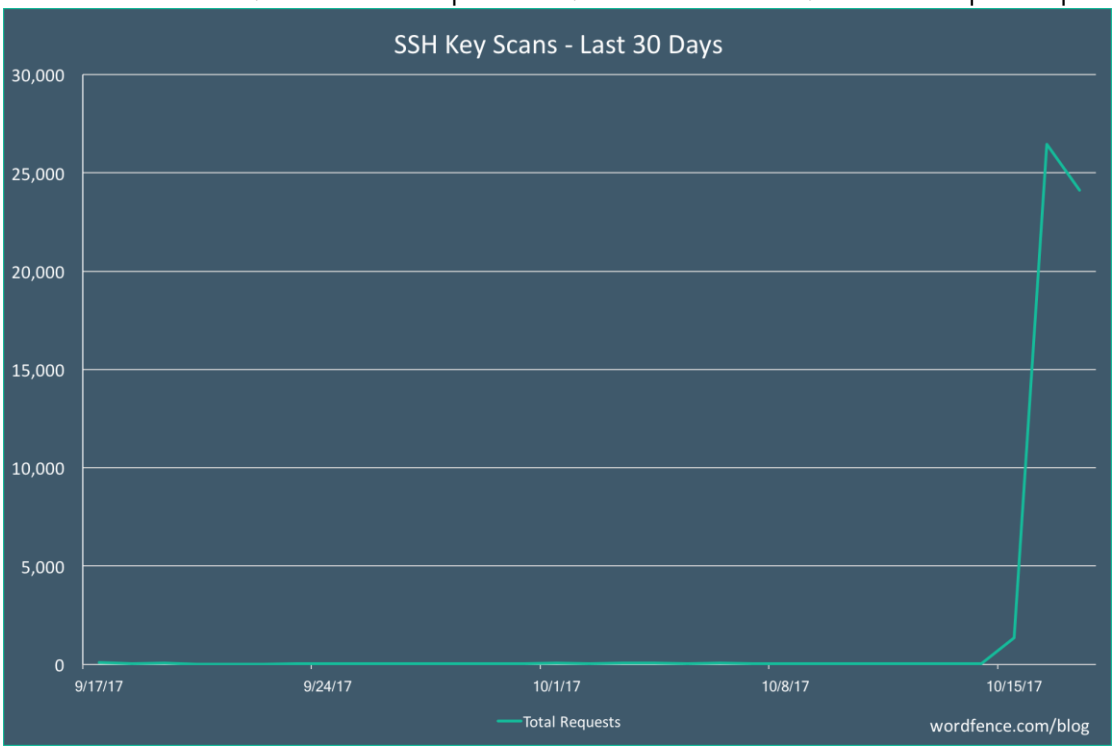
סורק אבטחה של WORDPRESS מאת Wordfence:

- סורק תוכנות זדוניות בודק קבצי ליבה, ערכות נושא ותוספים עבור תוכנות זדוניות, כתובות URL רעות, דלתות אחוריות, ספאם SEO, הפניות מחדש זדוניות והזרקות קוד.
- עדכוני חתימה של תוכנות זדוניות בזמן אמת באמצעות עדכון האיום.
- משווה את קבצי הליבה, ערכות הנושא והתוספים שלך עם מה שנמצא במאגר WordPress.org, בודק את תקינותם ומדווח לך על כל שינוי.
- מתקן קבצים שהשתנו על ידי החלפתם בגרסה מקורית וטהורה. מחק כל קבצים שאינם שייכים בקלות לממשק Wordfence.

- בודק את האתר שלך לאיתור פרצות אבטחה ידועות ומתריע על כל בעיה. מתריע גם על בעיות אבטחה אפשריות כאשר תוסף נסגר או נזנח.
- בודק את בטיחות התוכן שלך על ידי סריקת תוכן קבצים, פוסטים והערות לאיתור כתובות אתרים מסוכנות ותוכן חשוד.
- בודק אם האתר או ה-IP שלך נחסמו ברשימת חסימות עקב פעילות זדונית, יצירת ספאם או בעיית אבטחה אחרת.

Wordfence רשם פעילות תוקף בזמן שהתוקף מנסה לסרוק מגוון נתיבים של ספריות קבצים כדי למצוא את המיקום של מפתחות SSH פרטיים נגישים שאינם מוגנים.

הגרף שלהלן מדגים את העלייה הפתאומית בתעבורת סריקת SSH שנלכדה על ידי Wordfence:



לדברי מארק מאנדר מ-Wordfence, העלייה בפעילות מעידה על כך שהתוקפים מצליחים לסרוק את האינטרנט אחר מפתחות SSH פרטיים חשופים ולאחר מכן להשתמש במפתחות אלה כדי להתחבר לשרתים. תוקפים גילו שיש טעות תפעולית באופן שבו משתמשים מטפלים במפתחות SSH. לפיכך, שחקנים זדוניים השקיעו יותר ויותר מאמצים בניצול חוסר ההבנה סביב ניהול מפתח SSH.
התוקפים משתמשים במכונות ובתוכניות הנקראות botnets כדי לסרוק את האינטרנט לאיתור מכונות לא מוגנות.
botnets - רשת בוט היא רשת של מחשבים נגועים בתוכנות זדוניות שנשלטות על ידי רועה בוט. ה-bot herder הוא האדם שמפעיל את תשתית ה-botnet ומשתמש במחשבים שנפגעו כדי להפעיל התקפות שנועדו לקרוס את הרשת של המטרה, להחדיר תוכנות זדוניות, לאסוף אישורים או לבצע משימות עתירות מעבד

במאמר מתוארת הדרך שלהם להתמודד עם ההתקפה:

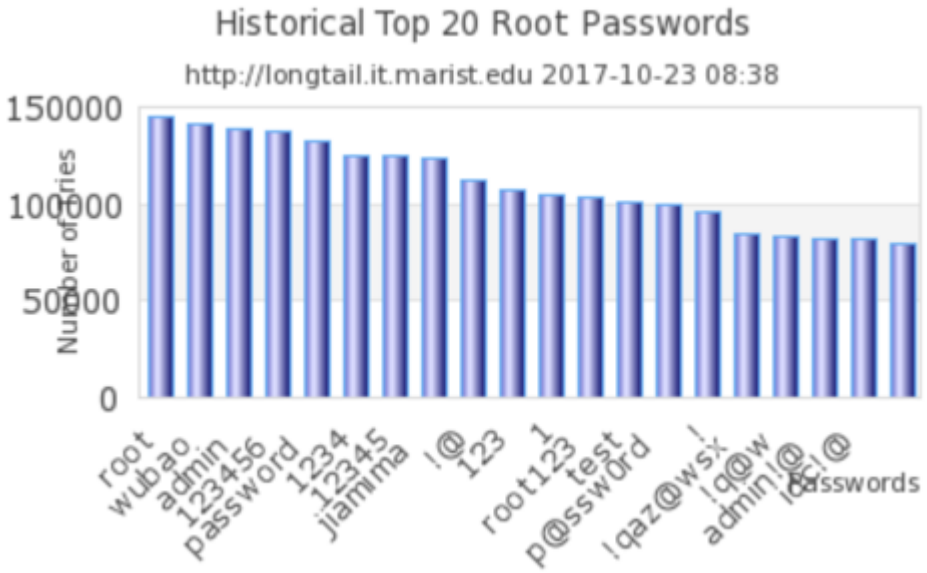
<u>שלב ראשון</u>, המלכודת מונחת-
Honeypot הוא שרת שנראה לגיטימי לשחקנים זדוניים, אבל במציאות מטרת השרת הזה (honeypot) היא לאסוף מידע על דפוסי תקיפה של האקרים. מידע זה עזר להם להבין טוב יותר באילו

שיטות האקרים משתמשים לשרתים. במקרה של סיר הדבש של Marist, הנקרא LongTail, שרת OpenSSH שונה לתיעוד מידע על כל מי שמכוון לשרת. סיר הדבש שמר מידע כגון שם המשתמש, הסיסמה וכתובת ה-IP שבה השתמש התוקף כדי לנסות להיכנס לשרת.

הטריק עם סיר דבש הוא לגרום לשרת להיראות בדיוק כמו שרת רגיל. אם התוקפים יבחינו במשהו חריג בשרת, הם יחשדו שמדובר במלכודת או בסיר דבש ויעברו ליעד הבא. The Marist LongTail honeypot מטעה תוקפים על ידי שימוש בקוד OpenSSH האמיתי עם שינויים קטנים והפעלת קוד זה כשרת ה-SSH בשרת ה-honeypot. סיר הדבש נראה בדיוק כמו שרת OpenSSH אמיתי מכיוון שהוא משתמש בקוד OpenSSH בפועל.

רוב ההתקפות על סיר הדבש של Marist הן SSH brute force, כלומר התוקף פשוט ניחש סיסמאות משתמש נפוצות עד שהצליחו. הרוב המכריע של התקיפות הגיע מסין. אחת הסיבות העיקריות לכך היא קבוצות פושעי סייבר הדומות ל-SSHPsychos המבוססות בסין. קבוצה זו כל כך פעילה בהתקפות הBrute Force שלהן שלפעמים הן מהוות עד 35% מכלל תעבורת SSH באינטרנט.

<u>שלב שני,</u>
סיסמאות רגישות להתקפות Brute Force ממספר סיבות, אך אחת הסיבות העיקריות היא שאנשים משתמשים בסיסמאות לא מאובטחות או קלות לניחוש. פרויקט LongTail מוכיח זאת על ידי איסוף הסיסמאות הנפוצות ביותר שניסו תוקפים. הסיסמאות הנפוצות ביותר שנוסות על ה-root (חשבון הניהול הפריבילגי של prizedv) הן גרועות בערך כפי שהיית מצפה, אבל עם כמה הפתעות: root, wubao, admin, 123456, password, 1234 ו-Wubao. jiamima מתורגם מסינית ל"ללא הגנה" או "ללא סיסמה" ו-jiamima מתרגם ל"מפתח הצפנה" או "סיסמה". כאמור, תוקפים מנסים את השיטות הללו כי הן עובדות.
20 סיסמאות השורש הנפוצות ביותר שהתגלו:



<u>שלב שלישי,</u>
הם חשדו שתוקפים מנסים לנצל את מפתחות SSH באותו אופן. אלה יכולים להיות מפתחות SSH פרטיים שהוכפלו והופצו בסביבה או מפתחות SSH פרטיים שנחשפו בטעות לרשת כפרטי דוח Wordfence. לאחר מכן הם החלו לאסוף נתונים על התקפות מפתח SSH. ביצעו כמה עדכונים לקוד Marist LongTail OpenSSH כדי ללכוד נתוני מפתח SSH כאשר תוקף מנסה לפרוץ את SSH באמצעות מפתח שנפרץ.

שלב רביעי,
בזמנו, סיר הדבש האחד שפורסם עם הקוד המעודכן ללכידת מפתחות SSH לא הראה התקפות. זה כנראה נבע מהעובדה שהוא נפרס רק על סיר דבש אחד ונחשב כמטרה בעלת ערך נמוך על ידי התוקפים. במילים אחרות, מכונה וירטואלית אוניברסיטאית לא שווה להתמקד בהתקפת מפתח SSH מתקדמת יותר כאשר Brute Force password פועלות וההחזר על ההשקעה עבור התוקף נמוך.

מפתחות SSH הם אישורים, בדיוק כמו סיסמאות, אבל יש הרבה הבדלים מסובכים בין מפתחות SSH לבין אישורים אחרים, כגון סיסמאות ותעודות. הבנת ההבדלים המורכבים הללו היא חיונית לניהול נכון ולאבטחת גישת מפתח SSH.

מיליוני מפתחות SSH פרטיים שאינם בשימוש שמחכים לניצול

צמד מפתחות SSH מכיל מפתח פרטי וציבורי כאחד. המפתח הפרטי מאוחסן במחשב שממנו אתה מתחבר והמפתח הציבורי נשמר במחשב המרוחק אליו אתה מתחבר. אסור לשתף מפתחות SSH פרטיים, כפי שהשם מרמז, לעולם לא להיות משותף או לחשוף לרשת הרחבה יותר מכיוון שמפתח זה מאפשר לכל אחד להתחזות לבעל המפתח הפרטי ולהתחבר לכל מכונה שהמפתח הפרטי מורשה להתחבר אליה. לעולם אל תציב אותם במקום שבו אנשים אחרים יכולים לקרוא אותם, לשלוח אותם באימייל, ל-FTP או להעביר אותם בצורה לא מאובטחת.

כדי לשלוט כראוי בגישה למפתח SSH שלך, תחילה עליך לגלות אילו מפתחות SSH יש לך, לאילו מכונות הם יכולים לגשת, ולקבוע אילו מפתחות SSH מפרים את תקנות התאימות על ידי היותם ישנים מדי או חלשים מדי. משם חשוב לעקוב אחר פעילות מפתח SSH לאורך זמן כדי לקבוע אילו מפתחות SSH אינם בשימוש או מיושנים. יש להסיר מפתחות שאינם בשימוש כדי לצמצם את משטח ההתקפה שלך וכדי להשיג ניצחונות מהירים.

SSH Vulnerabilities ופתרונן

אנו הולכים לפרט מספר נקודות תורפה בתוכנית ההתחברות ההצפנה SSH, נציין את התאריכים בהם נחשפו לפגיעות , והדרך לפתור אותן.
גרסאות מיושנות של ssh עשויות לאפשר למשתמש זדוני להיכנס כמשתמש אחר, להכניס פקודות שרירותיות להפעלה, או לקבל גישת שורש מרחוק לשרת ssh.

Secure Shell, או ssh, היא תוכנה המשמשת לכניסה למחשב אחר דרך רשת, ביצוע פקודות במחשב מרוחק והעברת קבצים ממחשב אחד לאחר. הוא מספק אימות חזק ותקשורת מאובטחת על פני ערוצי תקשורת לא מאובטחים. משתמש עם גישת "root" למכונות מסוימות ברשת, או גישה פיזית לרשת עצמה, עשוי לקבל גישה לא מורשית למערכות על ידי ניצול פגיעויות שונות שנמצאות בפקודות r" BSD". כמו כן, ייתכן שמשתמש זדוני יוכל לרשום את כל התעבורה אל מערכת היעד וממנה, כולל הקשות וסיסמאות. למערכת X Window יש גם מספר פגיעויות שעלולות להיות מנוצלות על ידי האקרים. השימוש ב-ssh עוזר לתקן את הפגיעויות הללו. באופן ספציפי, ssh מגן מפני התקפות אלה: זיוף IP (כאשר הזיוף נמצא על מארח מרוחק או מקומי), ניתוב מקור IP, זיוף DNS, יירוט של סיסמאות/נתונים ברורים והתקפות המבוססות על האזנה לנתוני אימות X וחיבורים מזויפים ל שרת X11.

הבעיות והפתרונות

1) פגיעות setsid של התחברות לא אינטראקטיבית
בכל פעם שמשתמש מרוחק נכנס באופן אינטראקטיבי לשרת באמצעות SSH, הפונקציה setsid משמשת כדי לשנות את תהליך הצאצא שנוצר להרשאות המשתמש. עם זאת, SSH תומך גם בחיבורים לא אינטראקטיביים, ובמקרה זה לא נקראת הפונקציה setsid, ותהליך הילד נשאר בקבוצת התהליך של תהליך המאסטר. לא ניתן היה לנצל את המצב הזה ישירות כדי להשיג הרשאות שורש, אך ייתכן שקיים ניצול אם יישום set-userid כלשהו במערכת מסתמך על שם הכניסה המוחזר על ידי פונקציית getlogin במקום מזהה המשתמש האמיתי או היעיל. פגיעות זו עלולה לאפשר גם יצירת רשומות יומן מזויפות בפלטפורמות מסוימות. ניצול יהיה אפשרי רק על ידי תוקף שכבר יש לו גישה לחשבון בשרת.
פתרון:
SSH Communications Security Secure Shell גרסאות 2.0.13 עד 3.2.1 מושפעות מפגיעות זו. כדי לפתור נעדכן לגרסאות 3.1.5 ו-3.2.2.

2) הצפת מאגר בזיהוי התקפת החדרה
CVE 2001-0144
מכיוון שפגיעות הכנסת ssh היא אינהרנטית בפרוטוקול SSH1, התווסף בלוק קוד ל-ssh1 שמזהה התקפת החדרה ומונע כל מנות שהן חלק מהתקפה. עם זאת, קוד זה הציג פגיעות חדשה הנגרמת על ידי מצב גלישה במשתנה מספר שלם שקובע את גודלו של מערך. זה מאפשר להקצות מערך בגודל אפס, המחזיר מצביע למרחב הכתובות של התוכנית עצמה. תוקף יכול לשלוח חבילה ארוכה בעלת מבנה מיוחד המנצל את המצב הזה, ובכך להפעיל קוד שרירותי בשרת.

SSH Communications Security גרסאות sshd 1.2.24 עד 1.2.31, וגרסאות OpenSSH לפני 2.3.0 פגיעות. בנוסף, גרסאות SSH Communications Security 2.x ו-x.3 פגיעות גם הן, אם SSH1 fallback מופעל, ומותקנת גם גרסה פגיעה של F-Secure 1.3.x. sshd 1.x ו-OSSH 1.5.7 ומטה גם הם פגיעים.

פתרון:
גרסאות ssh לפני 1.2.24 אינן פגיעות למתקפה זו, אך הן פגיעות למתקפה של ההכנסה.

3) פגיעות sshd AllowedAuthentications
23/05/02
Authentication-אימות מאפשר לארגונים לשמור על אבטחת הרשתות שלהם על ידי מתן רשות רק למשתמשים או תהליכים מאומתים לקבל גישה למשאבים המוגנים שלהם. זה עשוי לכלול מערכות מחשב, רשתות, מסדי נתונים, אתרי אינטרנט ויישומים או שירותים אחרים מבוססי רשת.
בסביבות שבהן הערך AllowedAuthentications באפשרויות התצורה (עבור SSH Communications Security SSH Secure Shell לשרתים ו-SSH Secure Shell עבור Windows Servers גרסאות 3.0 עד 3.1.1) השמט את מילת המפתח "Password" כאפשרות אימות, גרסת פרוטוקול מעטפת מאובטחת כלשהי ייתכן ש-2 לקוחות יוכלו לעקוף את התצורה כדי להשתמש באימות סיסמה. הדבר עלול להוביל למצב בו מנהל מערכת מתכוון להשתמש בשיטות אימות חזקות יותר (למשל, SecurID או אישורים דיגיטליים) ואינו אוכף דרישות לסיסמאות חזקות, בהנחה שאימות סיסמאות לא יתקיים כלל. פגיעות פוטנציאלית זו משפיעה על SSH Communications Security SSH Secure Shell עבור שרתים, SSH Secure Shell עבור תחנות עבודה (לקוח UNIX פועל במצב שרת) ו-SSH Secure Shell עבור Windows Servers גרסאות 3.0 עד 3.1.1. SSH Secure Shell עבור תחנות עבודה לקוח Windows ו-SSH Secure Shell עבור מחשבי כף יד אינם מושפעים מפגיעות זו.

פתרון:
שימוש ב-RequiredAuthentications במקום AllowedAuthentications בתצורה כדי לעקוף את הבעיה, ושדרוג לגרסה 3.1.2 של התוכנה הרלוונטית.
AllowedAuthentications ו-RequiredAuthentications הן אפשרויות עבודה עם OpenSSH :
AllowedAuthentications publickey,password
האפשרות AllowedAuthentications מציינת באילו שיטות אימות מותר להשתמש. עם אפשרות זו מנהל המערכת יכול לאלץ משתמשים להשלים מספר אימותים לפני שהם נחשבים מאומתים.
RequiredAuthentications publickey,password
האפשרות RequiredAuthentications הקשורה ל- AllowedAuthentications, מציינת אילו שיטות אימות המשתמשים צריכים להשלים לפני המשך. פרמטר זה חייב להיות זהה לאפשרות AllowedAuthentications אחרת השרת ימנע חיבור בכל פעם.

4) פגיעויות מרובות ב-OpenSSH
26/6/02

- CVE 2002-0640 ,CVE 2002-0639
שתי פגיעויות הקשורות לגלישה במאגר, האחת במנגנון האימות של אתגר/תגובה ב-OpenSSH גרסאות 2.9.9 עד 3.3, והשנייה במנגנון אימות מקלדת אינטראקטיבית PAM ב-OpenSSH 2.3.1p1 עד 3.3, עשויות לאפשר לתוקף מרוחק לבצע פקודות שרירותיות .
פתרון:
ניתן לעקוף את הפגיעויות הללו על ידי הגדרת ChallengeResponseAuthentication ו-PAMAuthenticationViaKbdInt כ-no ב-sshd_config והפעלה מחדש של שירות sshd.

-כשמגדירים את ChallengeResponseAuthentication ל-No זה מגן מפני עקיפת ההגבלה ללא סיסמה עבור כניסה לשורש.

-מערכות המשתמשות במודולי PAM המשתמשות באימות מקלדת אינטראקטיבית (PAMAuthenticationViaKbdInt), עשויות להיות פגיעות לביצוע מרחוק של קוד.

- CVE 2002-0083
3/8/02

הצפת מאגר של בייט אחד בקוד הערוץ של OpenSSH 2.0 עד 3.0.2 עלולה לאפשר לתוקף מרוחק עם חשבון בשרת לקבל הרשאות שורש. לא ידוע אם פגיעות זו יכולה להיות מנוצלת גם על ידי תוקף ללא חשבון.

- CVE 2000-0525, CVE 2001-0872

13/12/01

אם OpenSSH 3.0.1 או גרסה מוקדמת יותר פועלת עם אפשרות UseLogin מופעלת, זה אפשרי עבור משתמש מרוחק המאמת באמצעות שיטות אימות מבוססות מפתח לשנות משתני סביבה שהועברו לתהליך הכניסה. זה יכול לאפשר למשתמש מרוחק לקבל הרשאות שורש לאחר כניסה מוצלחת על ידי הגדרת משתנה הסביבה LD_PRELOAD לספריית אובייקטים משותפים משלו המכילה קוד בעל מבנה מיוחד. פגיעות שנייה באפשרות UseLogin עלולה לאפשר לתוקף מקומי לבצע פקודות עם הרשאות בסיס עקב כשל בביטול הרשאות ב-OpenSSH לפני 2.1.1. ניתן לנצל את שתי הפגיעויות הללו רק אם האפשרות UseLogin מופעלת בקובץ תצורת השרת, ואם לתוקף כבר יש חשבון בשרת.

- CVE 2001-0816, CVE 2001-1380

30/11/01

בהתאם לסדר שבו ממוקמים המפתחות בקובץ authorized_keys2, ייתכן שתוקף מרוחק יעקוף הגבלות בקרת גישה של כתובות IP ב-OpenSSH לפני 2.9.9, ובכך יאפשר ללקוחות עם צמדי מפתחות מוגבלים לקבל גישה נוספת.

- CVE 2002-0575

24/4/02

גלישת מאגר קיימת ב-OpenSSH 3.1 ואילך אם הידור עם תמיכה ב-Kerberos/AFS ו-KerberosTgtPassing או AFSTokenPassing הופעלו ב-sshd_config. אפשרויות אלה אינן מופעלות כברירת מחדל. הצפת המאגר יכולה לאפשר לתוקף מרוחק לקבל גישה ב-OpenSSH 2.9.9 ואילך, או לאפשר לתוקף מקומי להשיג הרשאות מוגברות ב-OpenSSH 3.0 עד 3.1, אלא אם האפשרות UsePrivsep מופעלת.

ייתכן שתוקף מרוחק יבצע אימות אם Kerberos V מופעל ב-OpenSSH לפני 3.0.1.

פתרון עבור על הבעיות:
ניתן לתקן בעיות אלו על ידי שדרוג ל-OpenSSH 3.4 ומעלה.

5) פגיעות של אימות סיסמה SSH 3.0.0

CVE 2001-0553

מערכות יוניקס מאפשרות למנהל מערכת לנעול חשבון על ידי שינוי הסיסמה המוצפנת של החשבון למחרוזת כגון "*", "!!", או "NP" ב-/etc/shadow, /etc/passwd, או בכל מקום שבו מאוחסנות סיסמאות מוצפנות. . מערכות יוניקס רבות מגיעות עם מספר חשבונות ניהול

הנעולים כברירת מחדל. פגם בתהליך אימות הסיסמה ב-SSH 3.0.0 עלול לאפשר לתוקף מרוחק להיכנס לכל חשבון שנגעל בצורה זו באמצעות כל סיסמה או ללא סיסמה.

SSH Communications Security sshd 3.0.0 עבור Unix עם אימות סיסמה מופעל מושפע מפגיעות זו אם חשבונות כלשהם ננעלים באמצעות מחרוזת בת אחד או שניים בשדה הסיסמה המוצפנת. SSH עבור Windows, SSH עבור Unix מלבד גרסה 3.0.0 ושרתי SSH שאינם משתמשים באימות סיסמה אינם מושפעים.

התיקון לבעיה זו הוא שדרוג ל-SSH 3.0.1 ומעלה. לחלופין, ניתן ליישם אחת מהדרכים הבאות לעקיפת הבעיה:

השבת את אימות הסיסמה והשתמש במקום זאת בצורה אחרת של אימות

השתמש במילות המפתח AllowUsers, DenyUsers, AllowGroups ו-DenyGroups בקובץ /etc/ssh2/ssh2_config כדי לאפשר גישה רק לחשבונות עם סיסמאות חוקיות. הקלד " man ssh2_config" למידע נוסף.

אם יש לך את הפצת המקור עבור SSH, בקובץ lib/sshsession/sshunixuser.c ליד קו 953, לפני:

/*אימות מתקבל אם הסיסמאות המוצפנות זהות. */

הוסף את השורות:

```
if (strlen(correct_passwd) < 13)
return FALSE;
```

ולהרכיב מחדש.

6) הצפת מאגר ב-ssh עם RSAREF2
CVE 1999-0834
RSAREF2 הוא יישום של אלגוריתם RSA, המשמש את ssh לאימות והחלפת מפתחות. מצב גלישת מאגר ב-ssh יחד עם מצב גלישת מאגר ב-RSAREF2 יכולים לאפשר לתוקף מרוחק לבצע פקודות שרירותיות עם ההרשאות של שרת ה-ssh, שהוא בדרך כלל root. גרסאות ssh 1.2.27 ואילך אם הידור עם האפשרות --with-rsaref פגיעות. האפשרות --with-rsaref אינה ברירת המחדל, כך שאם אפשרות זו לא צוינה במפורש בעת הידור ssh, אז היא אינה פגיעה.

פתרון:
ניתן לתקן בעיה זו על ידי שדרוג ל-ssh-1.2.28. אם זה לא אפשרי, התקן את תיקון ssh ואת תיקון RSAref2. ראה CERT Advisory 99-15 למידע נוסף על תיקונים. ניתן להשתמש בפקודת התיקון של UNIX כדי להחיל תיקונים אלה. הערה: הידור מחדש של ssh ללא אפשרות --with-rsaref יתקן את הפגיעות, אך עשוי להוות הפרה של הגבלת זכויות היוצרים על RSA אם נעשה שימוש בארצות הברית. עיין בקובץ COPYING בהפצת ה-ssh שלך לפרטים נוספים.

7) שחזור מפתח הפעלה ssh
CVE 2001-0361
ssh משתמש בתקן הצפנת מפתח ציבורי PKCS#1_1.5 כדי להחליף את מפתח ההפעלה בין הלקוח לשרת כאשר הפעלה מתחילה. תוקף עם יכולת ללכוד את הפגישה המוצפנת מהרשת יוכל לשחזר את מפתח הפגישה, ובכך לפענח את הפגישה, על ידי ניצול פגם בתקן PKCS#1_1.5. למרות שפגיעות זו טבועה בפרוטוקול SSH1, היא בלתי ניתנת לביצוע במצבים רבים בשל המספר הרב של חיבורים לשרת שיידרשו במהלך אורך החיים של השעה של מפתח ההפעלה.
פתרון:
התקפה זו אינה אפשרית נגד OpenSSH עקב מגבלות על מספר החיבורים לשרת. זה גם לא אפשרי נגד ssh-2.x שפועל במצב תאימות SSH1. יש להחיל את התיקון שניתן בייעוץ CORE-SDI על ssh-1.x עד ssh-1.2.31.

8) פגיעות של הכנסת ssh
ssh משתמש באלגוריתם בדיקת יתירות מחזורית של 32 סיביות (CRC-32) כדי לוודא שחבילה מכילה נתונים לגיטימיים. אם נעשה שימוש במצבי צופן מסוימים, תוקף מרוחק יכול ליצור חבילת ssh שתפענח לטקסט פשוט שרירותי, וחולשה באלגוריתם CRC-32 עלולה לאפשר לתוקף לזייף סכום בדיקה חוקי כך שהחבילה תיראה לגיטימית. על ידי הכנסת מנות כאלה להפעלה קיימת, התוקף יכול לבצע פקודות שרירותיות במערכת. ssh גרסאות 1.2.23 ואילך יש פגיעות זו, כמו F-Secure גרסאות 1.3.4 ואילך. אם אינכם בטוחים באיזו גרסה אתם מריצים, הקלידו ssh -V במערכת, והיא תספר לכם איזו גרסה מותקנת.

פתרון:
שדרוג ssh לגרסה 1.2.25 ומעלה, או לגרסה F-Secure 1.3.5 ומעלה. משתמשי F-Secure עם חוזה תמיכה יכולים לקבל שדרוג מהקמעונאי המקומי שלהם.

9) פגיעות ssh-agent
CVE 1999-0013
CVE 1999-0248
חבילת ssh כוללת תוכנית בשם ssh-agent. ה-ssh-agent מנהל את מפתחות ה-RSA עבור תוכנית ssh, והוא משמש בעיקר כדי לעזור למשתמשים להימנע מהצורך להקליד את ביטוי הסיסמה שלהם בכל פעם שהם רוצים להשתמש ב-ssh, סיסמה או scp. כאשר מופעלת, תוכנית ssh-agent יוצרת ספריית מצב 700 בספריית /tmp, ולאחר מכן יוצרת שקע AF_UNIX בספרייה זו. מאוחר יותר, המשתמש יריץ תוכנית בשם ssh-add, אשר מוסיפה את המפתח הפרטי שלו לקבוצת המפתחות המנוהלת על ידי תוכנית ssh-agent. כאשר משתמש מעוניין להשתמש בתוכנית הדורשת אימות מפתח RSA, לקוח ssh מתחבר לשקע AF_UNIX ומבקש מתוכנית ssh-agent את המפתח המתאים.

הפגיעות נעוצה בעובדה שכאשר לקוח ssh מתחבר לשקע AF_UNIX, הוא פועל כמשתמש-על, או כ-root, ומבצע בדיקת הרשאות לא מספקת. זה מאפשר למשתמשים להערים על לקוחות ה-ssh שלהם להשתמש באישורים השייכים למשתמשים אחרים. במילים אחרות, כל משתמש שמשתמש באימות RSA ומשתמש בתוכנת ssh-agent עלול להשתמש באישורים שלהם באופן שגוי על ידי משתמש זדוני, אשר לאחר מכן עלול לגשת באופן שגוי לשירותים או לתוכניות במחשב מארח.

פגיעות זו משפיעה על גרסאות UNIX של ssh בלבד. באופן ספציפי, ssh עבור גירסאות UNIX 1.2.17 עד 1.2.21 פגיעות אם מותקנות עם הרשאות ברירת מחדל. גרסאות של ssh לפני 1.2.17 כפופות להתקפה שונה (אך דומה מאוד). בנוסף, תוכניות F-Secure ssh, לפני

גרסה 1.3.3, חשופות להתקפה זו. גרסה 1.1 של לקוח ssh מבוסס Windows, הנמכרת על ידי F-Secure Corporation, וגרסאות a1.0/1.0 של לקוח ssh של Macintosh אינם חשופים להתקפה זו. אם אינך בטוח באיזו גרסה או מותג של ssh אתה מפעיל, הקלד "ssh -V" בשורת הפקודה והמידע הזה יינתן לך על ידי המערכת. אם אינך בטוח אם הגרסה או המותג של ssh שלך פגיעים להתקפה מסוג זה, אנא צור קשר עם הספק המתאים.

פתרון:

עבור אלה המשתמשים בגרסאות הלא מסחריות של ssh עבור UNIX, פגיעות זו עשויה להיפתר בקלות. כל שעליך לעשות הוא לשדרג לגירסת SSH 1.2.26 ואילך. עבור אלה המשתמשים בתוכנת F-Secure ssh, גרסה 1.3.3 מתקנת את בעיית האבטחה הזו. עבור אלה המשתמשים בחבילת F-Secure ssh, ויש להם חוזה תמיכה, התיקון לפגיעות זו הוא שדרוג לגרסה 1.3.3, אותה ניתן להשיג מקמעונאי מקומי.

אם התיקונים שלעיל אינם מעשיים, או אם מנהלי מערכת מעוניינים להשתמש בתיקון זמני עד שניתן יהיה ליישם את ההחלטות שלעיל, ישנה פתרון לבעיה זו. הדרך לעקיפת הבעיה הזמנית היא שמנהלי מערכת מסירים את סיביות ה-setuid מהקובץ הבינארי של ssh. זה ימנע מהתקיפה לפעול, אבל גם ישבית סוג של אימות המתועד כ-rhosts-RSA. לדוגמה, אם הקובץ הבינארי ssh נמצא בספריית /usr/local/bin, הפקודה הבאה תסיר את סיבית setuid מהבינארי :

"chmod u-s /usr/local/bin/ssh".

Brute Force Attack

התקפות Brute Force הן אמצעי לקביעת שילוב של שם משתמש וסיסמה או אסימון גיבוב על מנת לקבל גישה בלתי מורשית לחשבון, קובץ או מידע מוגן אחר. מתקפת כוח אכזרי היא שיטת התקפה מבוססת ניסוי וטעיה הפועלת על ידי ניחוש אישורים, נתיבי קבצים או כתובות אתרים, באמצעות לוגיקה או הפעלת כל שילובי המקלדת האפשריים. תוקף בודק באופן שיטתי את כל הסיסמאות וביטויי הסיסמה האפשריים עד למציאת הנכונה. לחלופין, התוקף יכול לנסות לנחש את המפתח שנוצר בדרך כלל מהסיסמה באמצעות פונקציית גזירת מפתח. זה ידוע בתור חיפוש מפתח ממצה.

תוקפים משתמשים לעתים קרובות בתוכנות זדוניות וכלים אחרים כדי להפוך את התהליך של התקפות Brute Force לאוטומטיות על ידי הפצת ההתקפה על פני מגוון מיקומי מקור או מינוף תוכנות זדוניות כדי לתקוף חשבונות פנימיים מוגנים. לכלים נפוצים כגון Hydra, Chaos, CrackMapExec ו- PoshC2 יש פונקציות של Brute Force.

מחשבים אחרונים שיוצרו ב-10 השנים האחרונות יכולים לפצח סיסמה אלפאנומרית בת 8 תווים - רישיות ואותיות קטנות, מספרים ותווים מיוחדים - תוך כשעתיים. מחשבים כל כך מהירים שהם יכולים לפענח Brute Force תוך חודשים בלבד. סוגים אלה של התקפות Brute Force ידועות כחיפוש מפתח ממצה, שבו המחשב מנסה כל שילוב אפשרי של כל דמות אפשרית כדי למצוא את השילוב הנכון.

<u>ההגנה הטובה ביותר מפני התקפת Brute Force היא להבטיח שהסיסמאות שלך חזקות ככל האפשר, להאט את הזמן שלוקח להאקר לפרוץ ולהגדיל את הסבירות שיוותר וימשיך הלאה.</u>

לאחר השגת גישה, תוקף עשוי לקבל גישה למידע פיננסי, להפיץ תוכנות זדוניות או לחטוף את המערכת שלך. ישנן כמה נקודות כניסה פגיעות להתקפות Brute Force:

1. SMB/CIFS BRUTE FORCE ATTACK:
   SMB הוא פרוטוקול תקשורת הפועל בשכבת היישום ומשמש בעיקר כדי לספק גישה משותפת אל קבצים, מדפסות, יציאות טוריות ותקשורת בין מחשבים ברשת. הפרוטוקול מספק גם תקשורת בין תהליכית עם מנגנון הרשאות המאפשר את ירושתן. רוב השימוש של SMB הוא במחשבים המריצים חלונות, שבהם הוא ידוע לעיתים קרובות כ"שכנים ברשת."
   בשנת 1996 שינתה מיקרוסופט את שמו של הפרוטוקול ל־**CIFS** (Common Internet File System), והוסיפה עוד תכונות, כולל תמיכה בקישורים סימבוליים וקישורים קשיחים (ריבוי מיקומים לקובץ), הגדילה את גודל הקובץ המקסימלי, וכן הוסיפה ניסיון ראשוני לתמוך בקישוריות ישירה על יציאת TCP מספר 445 בלי כל התוספות של NetBIOS. מיקרוסופט הגישה גם כמה מפרטים חלקיים של הפרוטוקול בתור טיוטות לסטנדרט ל-IETF אך אלו פגו בינתיים.
   ההתקפה:
   חסימת הודעות שרת (SMB) ומערכת קבצי אינטרנט נפוצה (CIFS) הם פרוטוקולי שיתוף קבצים ברשת הנפוצים ביותר בשימוש על ידי Windows. שניהם עלולים להיות פגיעים להתקפות Brute Force. ברגע שתוקף מקבל גישה לחשבון משתמש הוא יכול לגשת לקבצים, לנוע לרוחב או לנסות להסלים הרשאות.
2. SSH BRUTE FORCE ATTACK
   Secure Shell הוא פרוטוקול לתקשורת מחשבים המאפשר ביצוע פעולות על מחשב מרוחק לאחר תהליך הזדהות. הוא נועד להחליף את rlogin, RSH ו־telnet ולאפשר תקשורת מאובטחת ומוצפנת בין שני מחשבים לא תלויים ברשתות לא מאובטחות. SSH פועל מעל TCP, והפורט הסטנדרטי שלו הוא 22.

ההתקפה:
SSH או Secure Shell הוא פרוטוקול רשת המאפשר תקשורת מוצפנת על פני רשתות לא מאובטחות. SSH משמש לכניסות מרחוק, ביצוע פקודות, העברת קבצים ועוד. התקפות SSH brute force מושגות לרוב על ידי תוקף שמנסה שם משתמש וסיסמה נפוצים באלפי שרתים עד שהם מוצאים התאמה.

3. DNS BRUTE FORCE ATTACK

Domain Name System הוא פרוטוקול המאפשר גישה לבסיס נתונים מבוזר, על מנת שיחידות קצה ברשת האינטרנט יוכלו לתרגם שמות תחום הנוחים יותר לשימוש אנושי טבעי לכתובות הנומריות האמיתיות אליהן הן יפנו בזמן ההתקשרות. באמצעות ה-DNS ניתן להציע שירותים מבוססי שם נוספים, כגון רישום של שרתי דואר.

ההתקפה:
במקום לנחש סיסמה או שם משתמש, התקפות Brute Force על DNS יכולות לזהות את כל תת-הדומיינים באתר. תוקפים משתמשים בסקריפטים ובכלים אחרים כדי לשלוח שאילתות בעלות מראה לגיטימי. התוקף יכול להשתמש בזה כדי למפות תת-דומיינים זמינים, שמות מארחים ורשומות DNS - הכל במטרה למפות רשת בחיפוש אחר נקודות תורפה.

4. RDP BRUTE FORCE ATTACK

RDP- שולחן עבודה מרוחק
בתחום המחשוב, המונח שולחן עבודה מרוחק מתייחס לתוכנה או לתכונה של מערכת הפעלה המאפשרת להפעיל מרחוק את סביבת שולחן העבודה של מחשב אישי במערכת אחת, תוך שהוא מוצג במערכת מכשיר לקוח נפרדת. ליישומי שולחן עבודה מרוחק יש תכונות משתנות.

ההתקפה:
התקפות Brute Force על RDP הן בעלות נמוכה וקלה יחסית לביצוע. למרות שסוג זה של התקפת כוח אכזרי רועש, הוא יכול להיות יעיל ביותר בשל השותפות של סיסמאות חלשות ושימוש חוזר. תוקף עשוי לבצע התקפת Brute Force על חשבונות RDP כדי למצוא סיסמאות חלשות או אישורי כניסה חוקיים. ברגע שתוקף ניגש לסיסמאות או לאישורי התחברות חוקיים, הוא יכול לפתוח בקלות מספר הפעלות RDP ממכשיר יחיד כדי לשלוט במכשירים רבים ברשת.

הגנה מפני התקפות Brute Force

- כדי להקשות על גילוי סיסמאות, מנהלי אבטחה ומנהלי IT צריכים לאכוף מדיניות סיסמאות קפדנית עם דרישות אורך ומורכבות מינימליות. יש לאפשר גם זיהוי מרובה גורמים, במידת האפשר.
- עבור חשבונות משתמש, השתמש במדיניות נעילה המגבילה את מספר ניסיונות ההתחברות הכושלים כדי למנוע ניחוש סיסמאות. ניתן להשתמש ב-Captchas ביישומי אינטרנט כדי למנוע ניסיונות אוטומטיים של Brute Force.
- ניתן לשפר את זיהוי התקפות Brute Force באמצעות פענוח. התקפות Brute Force מתרחשות לעתים קרובות על גבי פרוטוקולים מוצפנים כדי להסתיר. למשל ניסיונות Brute Force שכיחים נגד RDP, שאינו רושם ניסיונות כניסה שנכשלו. Brute Force נפוץ גם נגד כמה פרוטוקולים של Active Directory ומסד נתונים. מסיבה זו, חיוני שלכלי אבטחה יהיו יכולות פענוח עבור כל הפרוטוקולים המקובלים בתעשייה כמו TLS ופרוטוקולי Microsoft כגון Kerberos, MS-RPC, SMBv3 ועוד.
- הגדל את אורך הסיסמה: יותר תווים שווים ליותר זמן לפיצוח Brute Force
- הגדל את מורכבות הסיסמה: אפשרויות נוספות עבור כל דמות גם מגדילות את הזמן עד ל Brute Force
- הגבל ניסיונות התחברות: התקפות Brute Force מגדילות מונה של ניסיונות כניסה כושלים ברוב שירותי הספרייה - הגנה טובה מפני התקפות Brute Force היא לנעול משתמשים לאחר כמה ניסיונות כושלים, ובכך לבטל התקפת כוח אכזרי שמתבצעת

- הטמעת Captcha: Captcha היא מערכת נפוצה לאימות שאדם הוא בן אדם באתרים ויכולה לעצור התקפות Brute Force שמתבצעות
- השתמש באימות רב-גורמי: אימות רב-גורמי מוסיף שכבת אבטחה שנייה לכל ניסיון התחברות הדורש התערבות אנושית שיכולה לעצור מתקפת Brute Force להצליח

היסטוריה של Brute Force

בעוד שטכניקות של Brute Force ששימשו בשבירת קוד לפני המצאת המחשבים המודרניים, כמה מהתקפות ה-Brute Force המתועדות ביותר בעידן המודרני תועדו במאמר משנת 1977 על ידי הקריפטולוגים ויטפילד דיפי ומרטין הלמן.

אמנם לא אמצעי התקפה יעיל במיוחד, אבל התקפות Brute Force הן אחת משיטות ההתקפה הוותיקות והאמינות ביותר. שיטת התקפה זו נמצאת בשימוש נרחב כיום, עם עלייה במספר המקרים המדווחים בשנת 2020.

מה תוקפים יכולים להרוויח?

- גישה לנתונים אישיים
- גישה למערכת שלך עבור פעילות זדונית
- יכולת לערוך את האתר שלך ולהרוס את המוניטין שלך
- יכולת להפיץ תוכנות זדוניות
- הרווח ממודעות או נתוני פעילות

בונוס משימה מספר 9

מחשב קוונטי

מהו מחשוב קוונטי?

מחשוב קוונטי הוא טכנולוגיה שצומחת במהירות הרותמת את חוקי מכניקת הקוונטים כדי לפתור בעיות מורכבות מדי עבור מחשבים קלאסיים.

מחשבים קוונטיים הם מכונות המשתמשות בתכונות הפיזיקה הקוונטית כדי לאחסן נתונים ולבצע חישובים. זה יכול להיות יתרון ביותר עבור משימות מסוימות שבהן הם יכולים לעלות בהרבה על מחשבי העל הטובים ביותר שלנו

מחשבים קלאסיים, הכוללים סמארטפונים ומחשבים ניידים, מקודדים מידע ב"סיביות" בינאריות שיכולות להיות 0 או 1. במחשב קוונטי, יחידת הזיכרון הבסיסית היא ביט קוונטי או קיוביט**.

****Qubit** קיוביט המונח קיוביט (אנגלית: Qubit; סיבית קוונטית) משמש כיחידת מידה למידע קוונטי, וגם לתיאור אלמנט אחסון המידע הקטן ביותר במחשב קוונטי. זהו האנלוג הקוונטי של הביט בתורת המידע הקלאסית. במחשב קוונטי, קיוביט הוא מערכת קוונטית בעלת שני מצבים.

קוויביטים נעשים באמצעות מערכות פיזיקליות, כגון ספין של אלקטרון או כיוון פוטון. מערכות אלו יכולות להיות בסידורים רבים ושונים בבת אחת, תכונה המכונה סופרפוזיציה קוונטית. ניתן גם לקשר קוויביטים זה לזה באופן בלתי נפרד באמצעות תופעה הנקראת הסתבכות קוונטית. התוצאה היא שסדרה של קיוביטים יכולה לייצג דברים שונים בו זמנית.

לדוגמה, שמונה סיביות מספיקות למחשב קלאסי כדי לייצג כל מספר בין 0 ל-255. אבל שמונה קיוביטים מספיקים למחשב קוונטי לייצג כל מספר בין 0 ל-255 בו זמנית. כמה מאות קיוביטים סבוכים יספיקו כדי לייצג יותר מספרים ממה שיש אטומים ביקום.

למה אנחנו צריכים מחשבים קוונטיים?

כאשר מדענים ומהנדסים נתקלים בבעיות קשות, הם פונים למחשבי-על. אלו הם מחשבים קלאסיים גדולים מאוד, לרוב עם אלפי ליבות מעבד ו-GPU קלאסיות. עם זאת, אפילו מחשבי-על נאבקים לפתור סוגים מסוימים של בעיות.
אם מחשב-על נתקע, זה כנראה בגלל שהמכונה הקלאסית הגדולה התבקשה לפתור בעיה בדרגה גבוהה של מורכבות. כאשר מחשבים קלאסיים נכשלים, זה נובע לרוב ממורכבות.

בעיות מורכבות הן בעיות עם הרבה משתנים המקיימים אינטראקציה בדרכים מסובכות.

כמה מהיר מחשב קוונטי?

מחשוב קוונטי הוא דור חדש של טכנולוגיה הכוללת סוג מחשב מהיר פי 158 מיליון ממחשב העל המתוחכם ביותר שיש לנו בעולם כיום.

OpenSSH Moves to Prevent 'Capture Now, Decrypt Later' Attacks

(מאמר)

נכתב על ידי:Ryan Naraine

ריאן נאריין הוא עורך ב-SecurityWeek ומנחה סדרת הפודקאסטים הפופולרית של שיחות אבטחה. ראיין הוא אסטרטג אבטחת סייבר ותיק שבנה תוכניות מעורבות באבטחה במותגים עולמיים גדולים.

OpenSSH היא חבילת תוכנות מחשב המספקות גישה מרוחקת מאובטחת ומוצפנת על גבי רשת מחשבים על בסיס פרוטוקול SSH . חבילה זו היא אחת מהחבילות אשר באות להגן עם נתונים ממחשבים קוונטיים (עליהם פירטנו בעמוד הקודם).OpenSSH נחשב לכלי הקישוריות המוגדר כברירת מחדל עבור כניסה מרחוק עם פרוטוקול SSH. הוא משמש להצפנת כל התעבורה כדי למנוע ציתות וחטיפת חיבורים.

במהלך השנים פותחו גרסאות רבות אשר נועדו להצפין את הנתונים ממחשבים הקוונטים, הגרסה החדשה ביותר שקיימת היום כוללת ציוד חדש אשר נועדו למנוע התקפות. הגרסה החדשה דוגלת ב"ללכוד עכשיו, לפענח מאוחר יותר" .

ישנן אזהרות רבות על פי מומחי אבטחה שלמחשבים קוונטיים בקנה מידה גדול יהיה מספיק כוח סוס כדי לשבור את ההצפנה המודרנית, מה שאומר שזו תהיה טעות להניח שהנתונים המוגנים היום יישארו מאובטחים לשנים הבאות. ייתכן ששחקני איום כבר אוספים כמויות גדולות של נתונים מוצפנים בתקווה שיום אחד יוכלו לגשת אליהם.

"ההערכה היא שהאלגוריתם של NTRU מתנגד להתקפות המתאפשרות על ידי מחשבים קוונטיים עתידיים והוא משויך לחילופי המפתחות X25519 ECDH (ברירת המחדל הקודמת) כמעצור נגד כל חולשה ב-NTRU Prime שעשויה להתגלות בעתיד. השילוב מבטיח שה- בורסה היברידית מציעה אבטחה טובה לפחות כמו הסטטוס קוו", הסביר ראיין.

"אנחנו מבצעים את השינוי הזה עכשיו (כלומר לפני מחשבים קוונטיים רלוונטיים לקריפטוגרפיה) כדי למנוע התקפות "ללכוד עכשיו, לפענח מאוחר יותר" שבהן יריב שיכול להקליט ולאחסן טקסט צופן הפעלה SSH יוכל לפענח אותו פעם אחת במחשב קוונטי מתקדם מספיק זמין", הוסיף.

<u>למרות שהמחשבים הקוונטים העדכניים עדיין אינם מסוגלים לבסס יתרון משמעותי על פני מחשבים מסורתיים, הבשלת הטכנולוגיה בשנים הקרובות צפויה ליצור בעיות שונות מנקודת מבט של אבטחת סייבר.</u>

Protection Against Side-Channel Attacks Added to OpenSSH

נכתב על ידי:Eduard Kovacs

Eduard Kovacs הוא עורך תורם ב-SecurityWeek. הוא עבד כמורה ל-IT בתיכון במשך שנתיים לפני שהחל קריירה בעיתונאות ככתב חדשות האבטחה של Softpedia. אדוארד הוא בעל תואר ראשון באינפורמטיקה תעשייתית ותואר שני בטכניקות מחשב מיושמות בהנדסת חשמל.

OpenSSH היא חבילת תוכנות מחשב המספקות גישה מרוחקת מאובטחת ומוצפנת על גבי רשת מחשבים על בסיס פרוטוקול SSH.

כאשר אדם מגדיר SSH לאימות מפתח ציבורי, מפתחות פרטיים מאפשרים גישה לחשבונות. אם מפתח פרטי נפגע, תוקף יכול לבצע אימות לחשבונות שבהם המפתח הפרטי הוא מהימן. ל-OpenSSH יש יתרונות רבים, אך תוקפים יכולים לנצל פגיעות מסוימות בכדי לנסות לעקוף את כמות הניסיונות שלהם להכניס סיסמא ובכך לבצע תקיפה.

חוקר המכונה KingCope מצא:

- ששת ניסיונות האימות המותרים כברירת מחדל על ידי OpenSSH לפני סגירת החיבור יכולים להיות מובסים על ידי ניצול פגיעות המאפשרת לתוקף לבקש מספר רב של הנחיות לסיסמה.
- תוקף מרוחק יכול לנסות עד 10,000 סיסמאות.
  זמן הכנסת הסיסמאות: 2 דקות.
- הפגיעות קשורה למנגנון האימות האינטראקטיבי של המקלדת וניתן לנצל אותה דרך האפשרות KbdInteractiveDevices.
  KbdInteractiveDevices-מציין את רשימת השיטות לשימוש באימות אינטראקטיבי במקלדת. שמות שיטות מרובים חייבים להיות מופרדים בפסיקים. ברירת המחדל היא להשתמש ברשימה שצוינה בשרת. השיטות הזמינות משתנות בהתאם למה שהשרת תומך. עבור שרת OpenSSH, ייתכן שהוא אפס או יותר.

"החלק המכריע הוא שאם התוקף יבקש 10,000 מכשירים אינטראקטיביים עם מקלדת, OpenSSH יבצע את הבקשה בקלות ויהיה בתוך לולאה לקבלת סיסמאות עד שיחרוג מהמכשירים שצוינו", הסביר החוקר בפוסט בבלוג.

KingCope פרסם קוד הוכחת מושג (PoC) כדי להדגים את קיומה של הפגיעות ב-OpenSSH 6.9.

הפגיעות משפיעה על מערכות שבהן אימות אינטראקטיבי במקלדת מופעל.

תאגיד MITER הקצה היום את מזהה CVE-2015-5600 לפגיעות זו.

:CVE-2015-5600
הפונקציה kbdint_next_device ב-auth2-chall.c ב-sshd ב-OpenSSH עד 6.9 אינה מגבילה כראוי את העיבוד של מכשירים אינטראקטיביים עם מקלדת בתוך חיבור יחיד, מה שמקל על תוקפים מרוחקים לבצע התקפות brute force או לגרום למניעת שירות (צריכת מעבד) באמצעות רשימה ארוכה ומשכפלת באפשרות - ssh oKbdInteractiveDevices, כפי שמדגים לקוח שונה המספק סיסמה שונה לכל אלמנט pam ברשימה זו.

"עד כמה שאנחנו יכולים לדעת, מהות הפגיעות היא שהלקוח לא אמור להיות מסוגל לציין מספר גדול באופן שרירותי של KbdInteractiveDevices ולזכות שהשרת ישתף פעולה", אמר צוות ההקצאה של MITRE.

המלצות להפחתת התקיפה מפי המומחים:

- הגבלת גישה ל-SSH בחומת האש
  כיצד להגביל גישת SSH לפי כתובות IP באמצעות חומת אש?
  ייתכן שתרצה ליצור SSH בשרת שלך או בקבוצה של שרתים שיהיה נגיש רק מרשתות IP מסוימות. כדי לעשות זאת, אתה יכול להגדיר כללי חומת אש מבוססי מארח באמצעות iptables, nftables, ufw וחומת אש. אתה יכול גם להשתמש בשירות חומת האש מבוססת הרשת המסופק על ידי Servers.com.
- השבתת אימות סיסמאות לחשבון השורש
- הפחתת התקפות brute force באמצעות מערכות זיהוי חדירה (IDS)
  IDS -מערכת לגילוי חדירות (Intrusion detection system)
  מערכת לגילוי חדירות היא התקן או תוכנה המנטרת את המערכת או תעבורת הרשת, חושפת ומתריעה על פעולות חשודות, ניסיונות גישה בלתי מורשים, התקפות על הרשת או כל פעולה אשר מנוגדת לחוקים שהוגדרו מראש
- שימוש בסיסמאות חזקות

רשימת מקורות

Videos we've watched:

- Beginners Guide To SSH
https://www.youtube.com/watch?v=qWKK_PNHnnA&t=291s&ab_channel=Tinkernut

- School Of Basics | What is SSH | How SSH works
https://www.youtube.com/watch?v=lRMAJwMQ0Vc&ab_channel=AutomationStepbyStep

- How Hackers Could Brute-Force SSH Credentials to Gain Access to Servers
https://www.youtube.com/watch?v=FKVsz_2IWJs&feature=emb_title
-

Sources:

- Wikipedia – Secure Shell
https://he.wikipedia.org/wiki/Secure_Shell

- Basics of SSH key authentication and management
https://www.manageengine.com/key-manager/information-center/what-is-ssh-key-management.html

- SSH Key Management
https://www.ssh.com/academy/iam/ssh-key-management

- How does proper SSH key management protect your network?
https://www.techtarget.com/searchsecurity/tip/How-does-proper-SSH-key-management-protect-your-network

- The top 6 SSH risks and how regular assessments cut danger
https://www.techtarget.com/searchsecurity/answer/SSH-security-risks-Assessment-and-remediation-planning

- https://www.techtarget.com/searchsecurity/tip/6-SSH-best-practices-to-protect-networks-from-attacks
https://www.techtarget.com/searchsecurity/tip/6-SSH-best-practices-to-protect-networks-from-attacks

- מאמר אקדמאי לרקע על הפרוטוקול

https://www.ssh.com/academy/ssh/protocol

- גרסאות SSH
https://www.omnisecu.com/tcpip/versions-of-ssh-protocol.php

- סכנות בשימוש SSH

https://www.venafi.com/blog/best-practices-ssh-key-management-what-are-your-ssh-security-risks

- יתרונות SSH

https://www.inmotionhosting.com/support/server/ssh/ssh-advantages/

https://www.venafi.com/blog/what-are-benefits-ssh-certificates

- התקפה ממשית של SSH
https://www.kb.cert.org/vuls/id/596827
- מאמר סריקת SSH
- https://www.ssh.com/blog/ssh-key-scan-attack-honeypot
- תקיפות ופתרונות
http://www.di-srv.unisa.it/~ads/corso-security/www/CORSO-0203/Scansione_servizi_rete/SAINT_DOCS/tutorials/vulnerability/SSH_vulnerabilities.html
- הצפנה SSH
https://www.omnisecu.com/tcpip/ssh-encryption-algorithms.php
- Four SSH Vulnerabilities You Should Not Ignore
https://www.cyberark.com/resources/blog/four-ssh-vulnerabilities-you-should-not-ignore
- Brute force
https://www.cmu.edu/iso/aware/be-aware/brute-force_ssh_attack.html
https://en.wikipedia.org/wiki/Brute-force_attack
https://www.varonis.com/blog/brute-force-attack
- מחשבים קוונטים:
https://www.newscientist.com/question/what-is-a-quantum-/computer

https://he.wikipedia.org/wiki/%D7%9E%D7%97%D7%A9%D7%91_%D7%A7%D7%95%D7%95%D7%A0%D7%98%D7%99

- https://www.analyticsinsight.net/10-difficult-problems-quantum-/computers-can-solve-easily

- KbdInteractiveDevices
https://man7.org/linux/man-pages/man5/ssh_config.5.html