**מטלות סיום : קורס הגנת פרוטוקולים  
SSH vulnerability**

**מגישים:**

יובל בר מעוז 314878877

רז אלבז ת"ז: 207276775

**חלק א'**

**תיאור המשימה ורשימת כוונות**

**בחירת המשימה עבור מטלת החקר:**

בחרנו במשימה 4.3: SSH vulnerability.

**חלק תאורטי:**

בחלק זה נתאר:

* סקירה רחבה על פרוטוקול SSH
* גרסאות פרוטוקול
* מפתחות SSH
* סקירת פגיעות ב SSH ברמה פנים ארגונית
* סקירת אבטחה בפרוטוקול SSH  
  <https://www.omnisecu.com/tcpip/ssh-encryption-algorithms.php>

**חלק מעשי:**

כתיבת כלי זיהוי או בדיקה/סריקה/תקיפה

**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:**הפרוטוקול משמש ברשתות ארגוניות עבור:

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

**כיצד פועל פרוטוקול SSH:**

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

בשלב השני, 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:

הלקוח יוזם חיבור על ידי פנייה לשרת

השרת שולח את הpublic key

לקוח SSH

לקוח SSH

שרת SSH

התחברות המשתמש למערכת ההפעלה המארחת של השרת

השרת והלקוח מנהלים משא ומתן על הפרמטרים ופותחים ערוץ מאובטח

**גרסאות פרוטוקול SSH**

יש בעיקר שתי גרסאות של פרוטוקול SSH. הגרסה הראשונית הייתה SSH-1, אשר שוחררה ביולי 1995. בשנת 2006, IETF (Internet Engineering Task Force) 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 ריחרוח 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-SSH1. SSH-2 מונע פרצות אבטחה רבות של SSH-1. SSH2 הוא הרבה יותר בטוח ויעיל מ-SSH-1. SSH-2 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. לניהול מערכות ותצורה אוטומטיות

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.

<https://www.manageengine.com/key-manager/information-center/what-is-ssh-key-management.html>

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: היכנס לשרת שממנו יש ליצור את החיבור המרוחק. צור זוג מפתחות באמצעות כלי ליצירת מפתחות SSH כמו PuTTYgen. אתה יכול גם להוסיף ביטוי סיסמה אופציונלי בזמן יצירת צמד המפתחות כדי לספק שכבת אבטחה נוספת.
* העתקת המפתחות הציבוריים למערכות מרוחקות: לאחר יצירת צמד המפתחות, העבר את המפתחות הציבוריים למערכות יעד מרוחקות. ודא שקובצי המפתח הציבורי ממוקמים תחת ספריית ~/.ssh/authorized\_keys בשרתים המרוחקים הדרושים. כאן, תצטרך לספק את ביטוי הסיסמה שנוצר בשלב הקודם.
* הפעל הפעלות SSH מרוחקות: לאחר שפרסת את המפתחות הציבוריים לשרתי היעד, תוכל 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.

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.

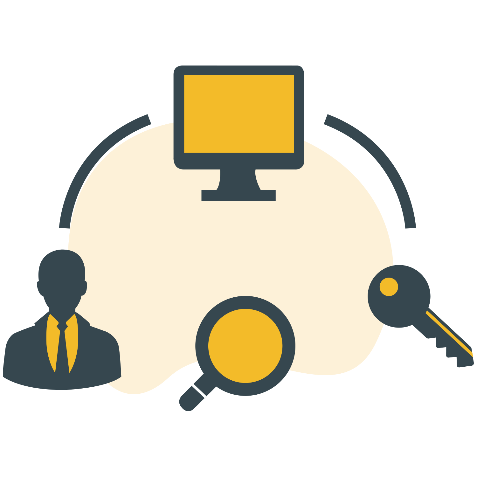
סיכוני אי ציות: ניהול מפתחות SSH חיוני גם עבור ארגונים כדי להציג ציות לתקנות מחייבות שונות בתעשייה כגון SOX, FISMA, PCI ו-HIPAA. אי שילוב מערכת ניהול מפתחות SSH נכונה עלול לגרום לחברות להיתקל בדרישות הציות ולעלות להן בקנסות אדירים.

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.



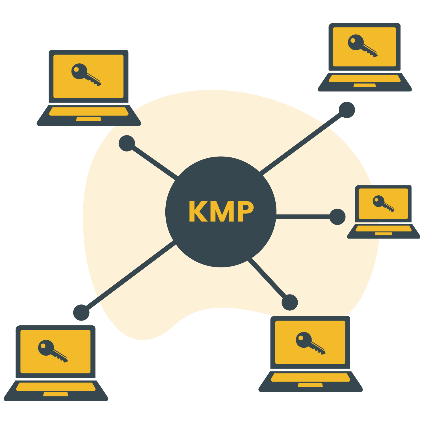
1. מפה את יחסי האמון

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.



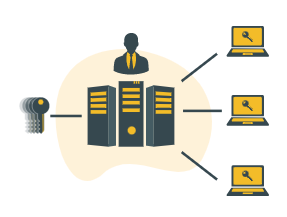
1. צור ופרוס צמדי מפתחות SSH טריים

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.



1. 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.



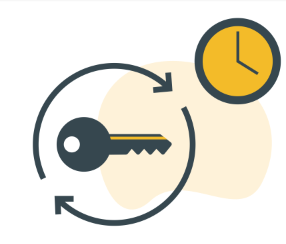
1. הטלת בקרות גישה מפורטות

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.



1. 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.



1. 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.

****

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>

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>