**חלק מעשי**

**מגישים:**

רז אלבז 207276775

יובל בר מעוז 314878877

**שפת כתיבה:**

Python 3.8

**דרישות:**Python גרסה 3.8

* Python pip
* Python module logging
* Python module paramiko
* Python module colorama
* Python module argparse
* Python module threading

**:מבוא**

במטלה זו התבקשנו לבצע מימוש של מתקפת SSH, המתקפה שבחרנו היא Brute Force Attack.  
התוכנית שלנו תרוץ על רשימה של שמות משתמשים ורשימה של סיסמאות שהודלפו, תנסה כל קומבינציה אפשרית עד אשר תצליח למצוא התאמה.  
לאחר שתצליח לבצע התאמה, "התוקף" ינצל את פתח הכניסה לארגון ויוכל להשיג את מטרותיו.

**שימוש בספריות חשובת במטלה:**  
 **pip**   
pip הוא מתקין החבילות עבור Python. אתה יכול להשתמש בו כדי להתקין חבילות מאינדקס החבילות של Python ואינדקסים אחרים.

**The Logging Module**  
מודול הרישום ב-Python הוא מודול מוכן לשימוש וחזק שנועד לענות על הצרכים של מתחילים כמו גם צוותים ארגוניים. הוא משמש את רוב ספריות Python של צד שלישי, כך שתוכל לשלב את הודעות היומן שלך עם אלה מהספריות הללו כדי לייצר יומן הומוגניות עבור היישום שלך.

כאשר מודול הרישום מיובא, אתה יכול להשתמש במשהו שנקרא " logger" כדי לרשום הודעות שאתה רוצה לראות. כברירת מחדל, ישנן 5 רמות סטנדרטיות המציינות את חומרת האירועים. לכל אחד יש שיטה מתאימה שניתן להשתמש בה לתיעוד אירועים ברמת חומרה זו. הרמות המוגדרות, לפי סדר חומרת הגובר, הן כדלקמן:

* לנפות
* מידע
* אַזהָרָה
* שְׁגִיאָה
* קריטי

מודול הרישום מספק לך logger ברירת מחדל המאפשר לך להתחיל בלי צורך לבצע הגדרות רבות.

**Module paramiko**Paramiko הוא יישום של פרוטוקול SSHv2 ב- Python. Paramiko מספקת פונקציונליות של שרת לקוח. הספר מכסה רק את הפונקציונליות של הלקוח.  
מכיוון ש-Paramiko אינו חלק מספריית המודולים הסטנדרטית של Python, יש להתקין אותה.  
החיבור נוצר בדרך זו: ראשית, לקוח נוצר והגדרת תצורת לקוח, לאחר מכן חיבור מופעל ומוחזרת הפעלה אינטראקטיבית:

In **[2]:** client **=** paramiko**.**SSHClient**()**

In **[3]:** client**.**set\_missing\_host\_key\_policy**(**paramiko**.**AutoAddPolicy**())**

In **[4]:** client**.**connect**(**hostname**=**"192.168.100.1"**,** username**=**"cisco"**,** password**=**"cisco"**,**

**...:** look\_for\_keys**=False,** allow\_agent**=False)**

In **[5]:** ssh **=** client**.**invoke\_shell**()**

SSHClient היא מחלקה המייצגת חיבור לשרת SSH. הוא מבצע אימות לקוח. מחרוזת set\_missing\_host\_key\_policy היא אופציונלית, היא מציינת באיזו מדיניות להשתמש בעת חיבור לשרת שהמפתח שלו לא ידוע. מדיניות paramiko.AutoAddPolicy() הוסף אוטומטית שם מארח ומפתח חדשים לאובייקט HostKeys המקומי.

Method connect מתחבר לשרת SSH ומאמת את החיבור. פרמטרים:

* look\_for\_keys - כברירת מחדל paramiko מבצעת אימות מפתח. כדי להשבית זאת, שים את הדגל ב-False
* allow\_agent - paramiko יכול להתחבר לסוכן SSH מקומי. זה הכרחי כאשר עובדים עם מפתחות ומכיוון שבמקרה זה האימות נעשה על ידי התחברות/סיסמה, יש לבטל אותו.

לאחר ביצוע הפקודה הקודמת כבר יש חיבור לשרת. השיטה invoke\_shell מאפשרת להגדיר הפעלת SSH אינטראקטיבית עם השרת.

**Colorama**מודולים וספריות מובנים רבים קיימים ב-Python כדי להדפיס את הטקסט הצבעוני בטרמינל. ה-Colorama הוא אחד ממודולי Python המובנים להצגת הטקסט בצבעים שונים. הוא משמש כדי להפוך את הקוד לקריאה יותר. שלוש אפשרויות עיצוב זמינות במודול זה לצביעת טקסט. אלה הם Back, Fore ו- Style. ניתן לשנות את צבע הרקע או החזית של הטקסט ואת סגנון הטקסט באמצעות מודול זה. שימושים שונים במודול זה הוסברו במדריך זה.

**argparse - מנתח עבור אפשרויות שורת פקודה, ארגומנטים ותתי פקודות**

מודול argparse מקל על כתיבת ממשקי שורת פקודה ידידותיים למשתמש. התוכנית מגדירה אילו ארגומנטים היא דורשת, ו-argparse יבין כיצד לנתח אותם מתוך sys.argv. מודול argparse גם מייצר אוטומטית הודעות עזרה ושימוש ומוציא שגיאות כאשר משתמשים נותנים לתוכנית ארגומנטים לא חוקיים.

**Python module threading**

מודול זה בונה ממשקי השרשור ברמה גבוהה יותר על גבי מודול \_thread ברמה נמוכה יותר. לסנכרון מסופקים מנעולים פשוטים (הנקראים גם mutexes או סמפורים בינאריים). מודול השרשור מספק ממשק API לשרשור קל יותר לשימוש וברמה גבוהה יותר שנבנה על גבי מודול זה.

**הרצה(ב Linux):**

1. התקן את המודולים הדרושים במטלה
2. ***Brute force***  של משתמש יחיד:

python3 brute\_force\_ssh.py -i **IP** -u **USERNAME** -P leaked\_lists/passwords.txt

1. ***Brute force***  של כל המשתמשים:

python3 brute\_force\_ssh.py -i **IP** -U leaked\_lists/usernames.txt -P leaked\_lists/passwords.txt

**תמונות מההרצה:**

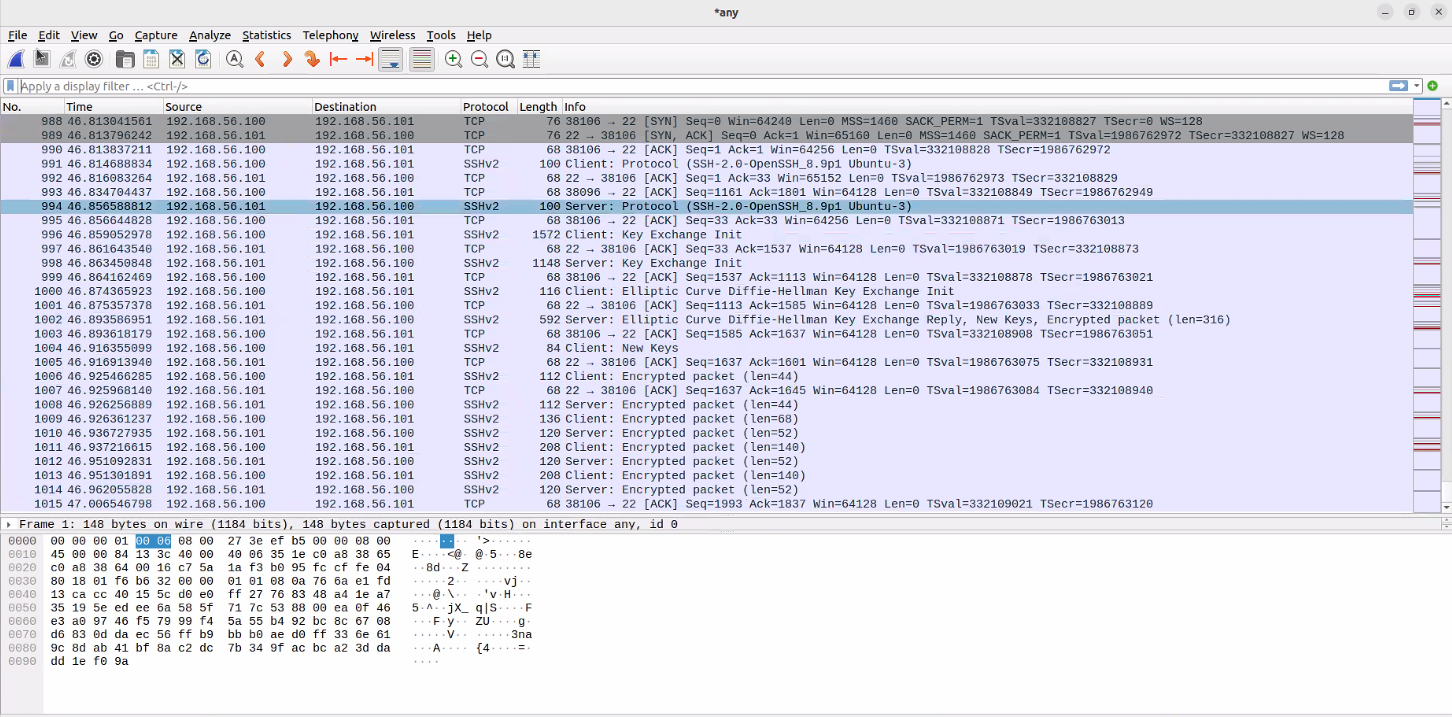
תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי של משתמש יחיד ***Brute force***

כפי שניתן לראות אנו מריצים את התוכנית עם שם משתמש של יוזר ספיצי וממתינים שהתוכנית תבצע את התקיפה ותמצא עבורנו את הסיסמא המתאימה לאותו שם משתמש. התוכנית מבצעת התקיפה על ידי מעבר על רשימת הסיסמאות אשר נדלפה, תנסה לבצע חיבור של השם משתמש הספציפי שהכנסנו וכל סיסמא אפשרית, במידה וקיימת התאמה נראה את ההדפסה "Connected" ואם אין נראה את ההדפסה "Rejected ".  
לאחר שנמצאה הסיסמא, יודפס לנו שם המשתמש והסיסמא המתאימים   


ואז התוקף יוכל להקליד את הסיסמא שנמצאה, להתחבר ולהשיג את המטרה שלו:



**הרצה בWireshark:**

כפי שניתן לראות התוכנית שלנו מבצעת את התקיפה בשימוש בפרוטוקול ssh.

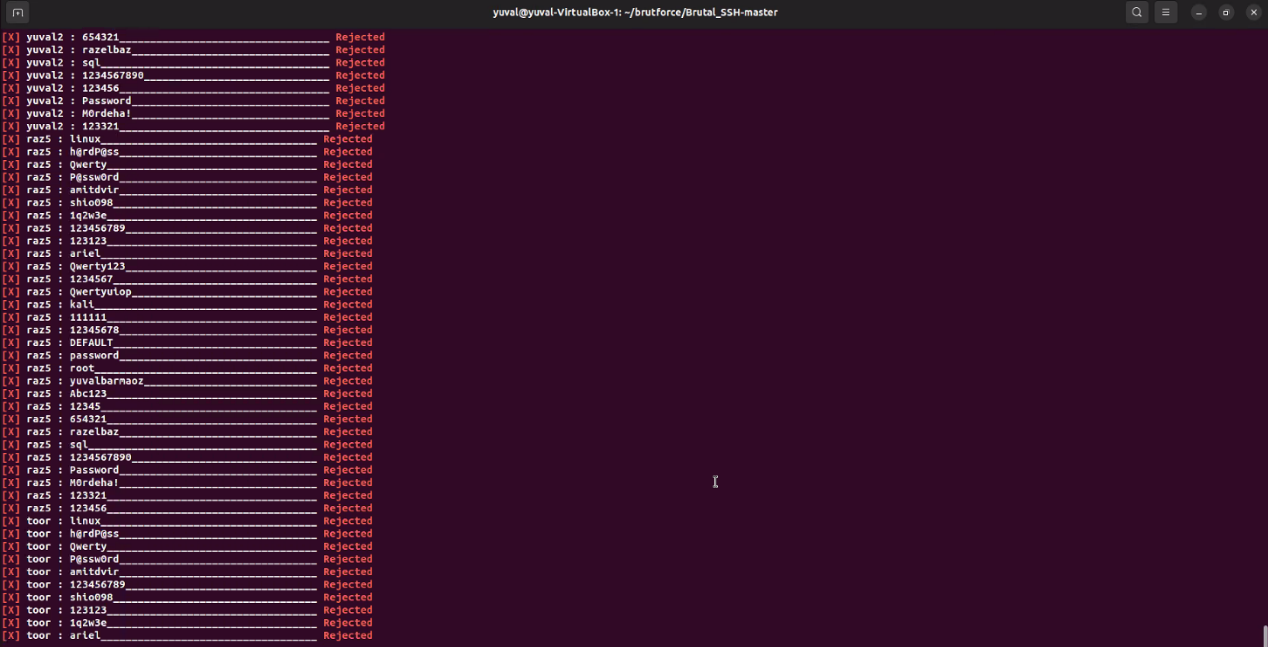
**ניסיון תקיפה כושל ( אין התאמה בין שם משתמש לסיסמא כלשהי ברשימה):**

**ניסיון תקיפה כושל (ה-IP ):**

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

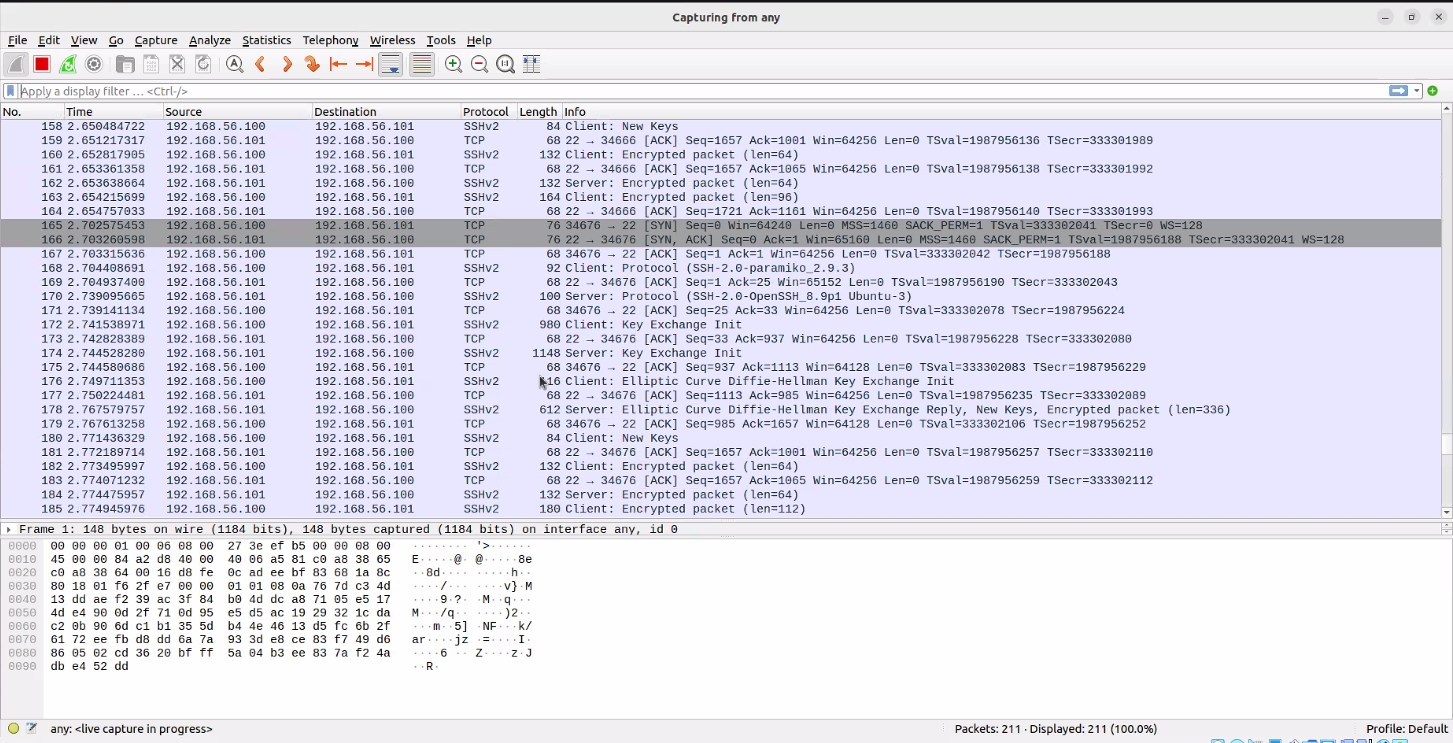
***Brute force***  של כל המשתמשים

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

מבוסס על אותו רעיון כמו על הרצה יחידה, רק שהפעם התוכנית תנסה כל קומבינציה אפשרית של שם משתמש מהרשימה שנדלפה עם כל סיסמא מרשימת הסיסמאות שדלפו.

**הרצה בWireshark:**

****