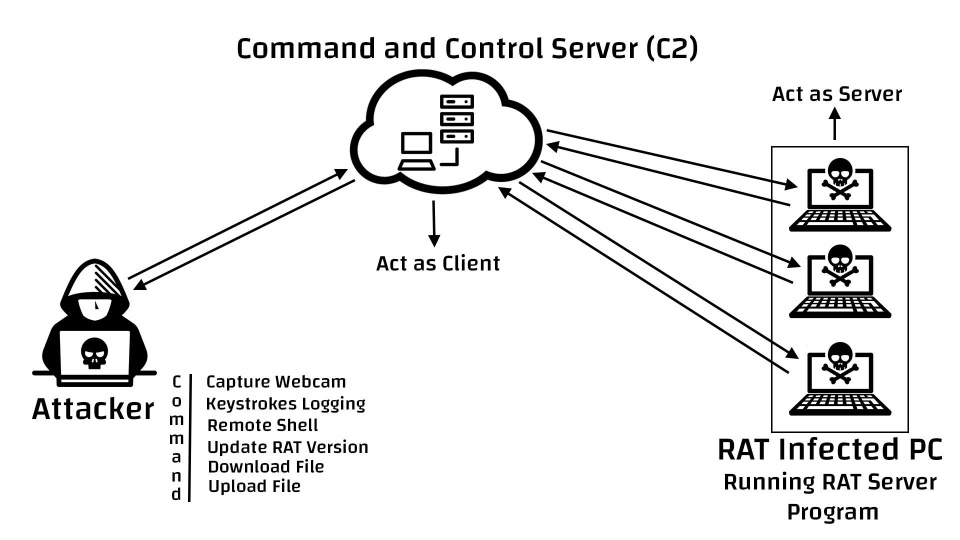
הקדמה

קצת רקע על מה זה Command and Control, היא מערכת שאליו יתחברו עמדות ש"הודבקו" ב-malware כלשהו והmalware מייצר backdoor כך שממנו יהיה ניתן לשלוט על העמדות.

לצורך המחשה אפשר להסתכל על התמונה:

השרת יתמוך בריבוי לקוחות ויספק ממשק למשתמש להזנת פקודות לביצוע , הלקוח ירוץ על העמדה ש"הותקפה" וירוץ תמיד עם אתחול המערכת על מנת ליצור PERSISTANCE .

הלקוח יבקש משימות לביצוע מהשרת וישלח את התוצאות חזרה אל השרת.

מימוש:

נשתמש בשפת Python ובעזרת Flask נממש את השרת C2 , כמו כן ממשק web שדרכו יוכל התוקף לשלוח משימות אל הלקוחות.

נממש גם תוכנה שתרוץ בצד הלקוח ותבצע את המשימות.

תוכנה שרצה בצד הלקוח:

1. התוכנה תשלח בקשות לשרת בכל פרק זמן שיוגדר מראש ותבדוק האם ממתינות לה משימות לביצוע.
2. ביצוע המשימה.
3. שליחת התוצאות לשרת.
4. Persist

תוכנה שירוץ בצד של שרת :

1. הוספת/הסרה של לקוחות.
2. ממשק UI שמקבל פקודות מהיוזר.
3. הכנסת הפקודות לתוך רשימה משימות לביצוע של הלקוחות.
4. האזנה לבקשות של לקוחות ובדיקה האם קיימת משימת בשבילם.
5. לאחר שליחת משימה לביצוע , השרת ימתין לתוצאה מהלקוח.
6. טיפול במספר לקוחות במקביל.
7. תמיכה במע' הפעלה לינוקס ווינדווס.
8. הצפנת תקשורת בין לוקח ושרת ע''י שימוש ב-HTTPS.

פונקציונאליות:

1. העלאת קבצים.
2. הורדת קבצים.
3. הרצת פקודה על מחשב לקוח.
4. קבלת SHELL אינטרקטיבי.

הסבר על הרכיבים:

1. ה-Server – הממשק ה-UI ממומש בעזרת flask, ושרת שמנהל את החיבורים של המשתמשים.

הממשק הUI בגדול מציג את הנתונים של השרת החיבורים, הגישה לשרת UI היא מוצפנת HTTPS שמשתמש ב-TLSv1, מכיוון שהשרת לא מונגש לעולם אז בוצע יצירה של self-signed certificate. השרת לא בהכרח יעבוד ב-HTTPS עם הסרטיפיקציות שנמצאים כאן. עבור כל משתמש יש חלון נפרד שניתן לשלוח פקודות למשתמשים. בנוסף השרת מנהל את החיבורים למשתמשים מבצע את השליחה וקבלה של הפקודות ב-TLS.

1. ה-Client – מתחבר בפורט 5555 לשרת, לאחר חיבור לשרת הקליינט והשרת מתחילים לתקשר ב-TLS. לקליינט. הקליינט יכול לשלוח ולקבל קבצים מהשרת שמתקבלים כקלט מהיוזר.

תרשים UML

שרת:

Diagram

Description automatically generated

קליינט:

