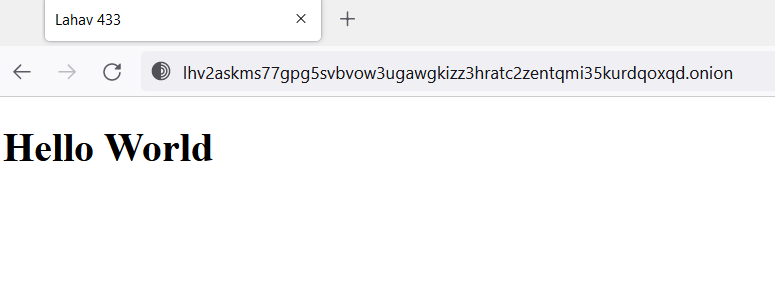
1. ***הסבר כיצד עובדת רשת TOR וכיצד ניתן לשמור על אנונימיות באמצעות הרשת.***

רשת התור היא על פניו דפדפן המתנהג רגיל כמו כל שאר הדפדפנים אשר ההבדל העיקרי טמון בשמירת האנונימיות של המשתמש .  
באופן רגיל , תהליך יצירת התקשורת עם שרת חיצוני מתנהלת באמצעות קליינט (הלקוח) אשר משתמש בדפדפן קיים , ויוצר בקשת TCP (ייתכן גם פרוטוקול אחר) אל מול שרת אינטרנט , וזה באמצעות אפליקציית הדפדפן, במקרה הנ"ל מטרת הדפדפן היא פשוט לנהל מאחורי הקלעים את תהליך יצירת הקשר בין הלקוח לשרת והצגת המידע באפליקציית הדפדפן. בתהליך זה השרת מכיר את ה IP של הלקוח , ובאמצעות שיטות שונות ניתן לאתר את מקום הימצאו של הלקוח ולמעשה לאמת את זהות המשתמש .  
דפדפן ה TOR שונה בהתנהגותו מדפדפן רגיל בכך שהוא שומר על אנונימיות המשתמשים שלו באמצעות מנגנון שרבים קוראים לו “onion router” , במקום ליצור קשר ישיר בין הלקוח לשרת – tor "מחביא" את המידע של הלקוח באמצעות ניתוב המידע דרך שלושה צמתים רנדומליים ברשת , כאשר בכל צומת המידע מוצפן על ידי מפתח נוסף ובנוסף בעצם העברת המידע בין הצמתים , כל צומת מודע למי שלח לו את המידע ולאן הוא צריך להעביר , אבל אף אחד לא יודע את המשך שרשרת הצמתים , ובכך IP המקור הוסתר, והשרת אינו יודע מי הלקוח אשר ביצע את הבקשה ובכך הלקוח שמר על האנונימיות שלו.  
בנוסף, דפדפן tor מונע (ניתן להשבית יכולת זאת) הפעלת סקריפטים בדפדפן שייתכן ויכולים להחדיר נוזקה ללקוח כדי למצוא את זהותו .

במובן הטכני, ללקוח אשר משתמש ברשת הtor יש שלושה מפתחות הצפנה , מפתח לכל צומת שהוא משתמש בה ליצירת תקשורת עם השרת . כאשר הלקוח מעביר בקשה לשרת הוא מצפין את המידע באמצעות שלושת המפתחות לפי סדר הצמתים בהעברה(אם נצייר את פעולת ההצפנה זה ידמה לצורת הבצל , כי יש שכבות שונות של הצפנה) , כאשר המידע מגיע לצומת הוא עובר פענוח ובצורה זו כל צומת יודעת לאן להעביר את ההודעה , תהליך זה מתבצע עד שהמידע פוענח לגמרי וניתן לדעת לאיזה שרת להעביר את הבקשה כיעד סופי .   
לאחר שהשרת מבצע את הבקשה והוא רוצה להעביר את התשובה בחזרה ללקוח – המידע עובר בחזרה באותו סדר של הצמתים (כי כל צומת מכיר רק את הצומת הבא וזה שלפניו , והשרת מכיר רק את הצומת שממנו קיבל את המידע) , ובכל צומת המידע מוצפן מחדש עד שמגיע בחזרה אל הלקוח שמפענח עם המפתחות שלו את כל שכבות ההצפנה .

למרות שלמשתמש הפשוט tor נראה כמו דפדפן , רבים מתייחסים אליו כאל רשת עצמאית מהסיבה שהוא מאפשר גישה לאתרים שלא קיימים ברשת הרגילה (Dark Web) , דוגמה כזו היא שהוא מאפשר גישה לאתרים מהדומיין .onion הסבר לזה יהיה בהמשך העבודה.

1. ***בנה אתר עם התוכן hello world, ברשת ה tor, אשר תחילת הכתובת תתחיל ב lhv***

******

ראשית הקמתי שרת Apache באמצעות XAMPP והגדרתי את הקבצים מחדש כך שלמעשה הוא יעלה תכנים מסוימים לכתובת המתאימה ברשת tor . כדי לייצר כתובת URL מתאימה -שתתחיל ב lhv , השתמשתי בסקריפט שנקרא mkp224o

**הכתובת לאתר היא**: http://lhv2askms77gpg5svbvow3ugawgkizz3hratc2zentqmi35kurdqoxqd.onion/

***בנוסף, הסבר כיצד עובד מנגנון הכתובות ברשת וכיצד בעצם ניגשים לאתר כלשהו.***

כדי לשמור על אנונימיות המשתמשים – רשת tor היא מבוזרת , אם היא הייתה ריכוזית אז זה אומר שהאנשים מתוך הארגון בעלי שליטה בכל נקודות התקשורת וההצפנה שהרשת משתמשת בה כדי להחביא ולהצפין את המידע , כלומר המידע למעשה חשוף ונגיש לאנשים מתוך הארגון ולמעשה ניתנים לפריצה ולחשיפה.  
מכיוון שהרשת מבוזרת , ניתן להבין שגם שאין ארגון כגון ה ICCAN שאחראי על מערכת DNS פנימית ברשת ומכך נובע כי לכתובות ב deep/dark web אין כתובות Ip , ולכן לא ניתן למצוא אותן דרך שאילתת DNS שמתקיימת בדפדפנים שונים.  
ומכיוון שאין שום קשר בין ה URL לכתובת Ip , החליטו ברשת tor לתת לאתרים אשר משתמשים בה שלושה פרטים : 1.מפתח פרטי 2. מפתח ציבורי 3. כתובת URL  
הכתובת URL היא למעשה כתובת מוצפנת המבוססת על בסיס המפתח הציבורי של השרת – וכאשר משתמשים במפתח הפרטי ניתן לפענח את הכתובת URL לכתובת לשרת שמאכלס את השירות onion המבוקש .

בעבר כתובת ה URL הייתה באורך 16 תווים אבל בעקבות שימוש בכל הכתובות החליטו להתחיל להשתמש בכתובות בגודל 56 תווים.

1. ***בהנחה וגוף מסוים בעל שליטה בשלושה צמתים של הרשת, ואדם מסוים משתמש בכל השלושה במסלולו אל היעד:*** 
   1. ***האם, ואם כן כיצד ניתן לדעת שמשתמש עובר דך כל שלושת הצמתים?***

כן. באופן עקרוני מספיק שאדם יהיה בעל שליטה בשני צמתי היציאה בלבד (יותר נכון לומר entry and exit nodes ) כדי שהוא יוכל לדעת מה התוכן העברה ומי היא כתובת היעד והלקוח . השליטה בשני צמתים אלו תספיק מכיוון שהשליטה בצומת הראשון מאפשרת לדעת את כתובת הלקוח והשליטה בצומת האחרון מאפשרת לדעת שההודעה אכן הגיעה לצד שרת (שהיא כבר לא מוצפנת) וכך גם נוכל לדעת במידה והפרוטוקול לא מוצפן את תוכן ההודעה וגם את כתובת הלקוח.

למרות זאת חשוב להבין שהיכולת לקשר בין פיסות התעבורה שניתן לקלוט באמצעות sniffing בצמתים השונים היא לא פעולה פשוטה ,והיא כן דורשת כוח מחשובי כדי לזהות שאכן ההודעות http בצומת אחד הן אותן ההודעות בצומת אחר .ניתן לבצע זאת מתוך ההבנה שלהודעות העוברות ברשת יש חותמת זמן , ואם נחפש חותמות זמן זהות בין צמתים שונים(וזו יכולה להיות משימה קשה כי כמובן שיש דילי בין הצמתים ועוברות ביניהם עוד הודעות רבות אחרות בין צמתים ,יעדים ומקורות שונים) אז נוכל אכן לאמת שההודעות אשר עברו בצומת אחת הן אותן ההודעות שעברו בצומת אחרת , כלומר אכן אימתנו את השימוש בכל שלושת הצמתים, ואף גילינו מי הלקוח ומה תוכן המידע שהועבר (במידה והפרוטוקול לא מוצפן)

* 1. ***האם ניתן לקבל את תוכן התעבורה שלו במידה ומדובר ב-http?***

כן. מכיוון שהודעות http אינן מוצפנות , אם נבצע sniffing על צומת ה output שגם מקבלת את ההודעות מהשרת כאשר הן עדיין לא מוצפנות – נוכל לדעת את תכני ההודעות , אך כאמור לא נדע למי היא מיועדת אם לא נבצע את מה שהסברתי בסעיף לעיל.

**שאלה שנייה – תכנות**

מטרת המשימה הינה בניית בוט בטלגרם אשר מקבל כתובת Ip או דומיין ומאפשר את שלוש האפשרויות הבאות:

1. קבלת שירותים פתוחים (open ports).
2. קבלת מערכת ההפעלה.
3. קבלת רשימת חולשות במידה ויש עבור השירותים הפתוחים.
4. קבלת כל האפשרויות לעיל בלחיצת כפתור אחת.

**קוד לgithub :** [**https://github.com/Nivarad/Nmap-Telegram-Bot**](https://github.com/Nivarad/Nmap-Telegram-Bot)