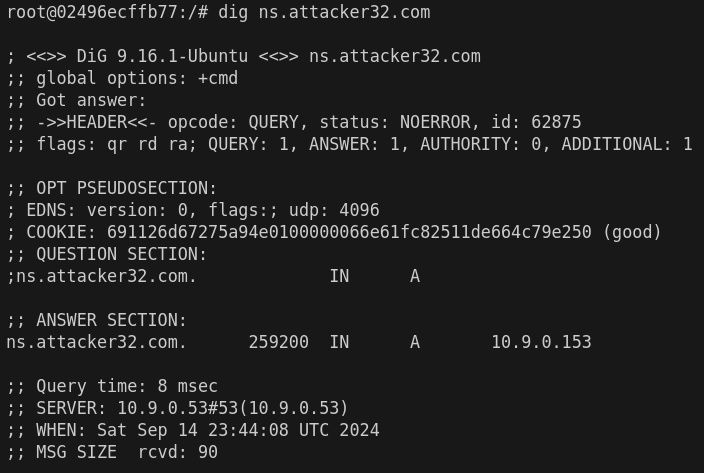
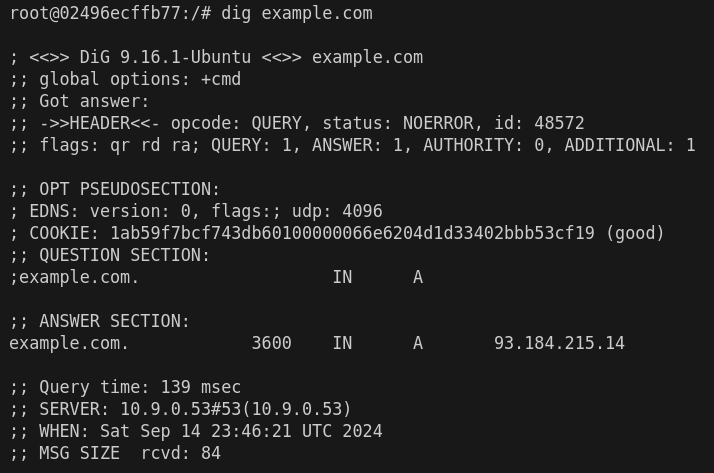
לאחר שהרמנו את כל ה-Container, נבדוק את מה שנדרש:

2.4:

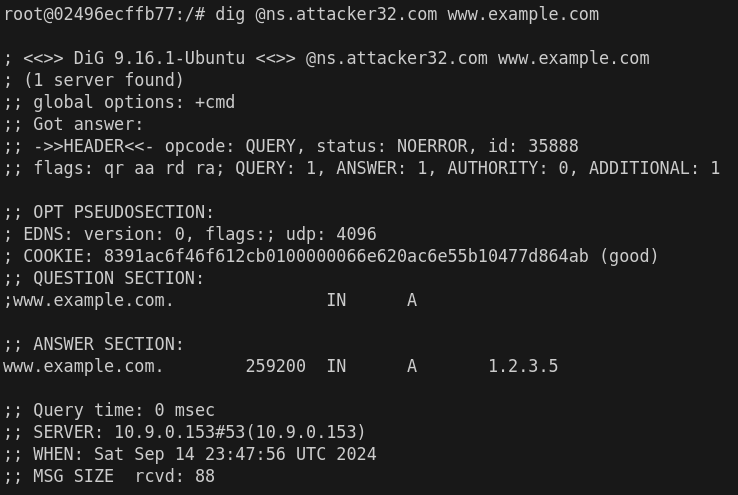
נבצע dig ns.attacker32.com ב-User ונקבל שכתובת ה-IP של ה-ns היא 10.9.0.153:



כעת, נריץ dig example.com (באופן רגיל) ונקבל את הכתובת הבאה:

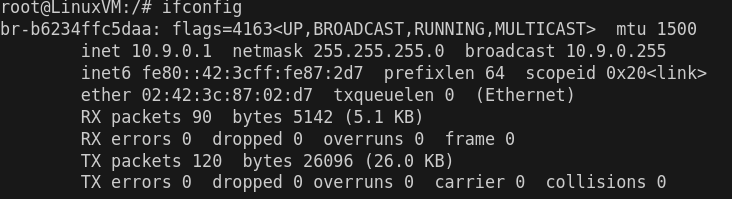


וכאשר נריץ את אותה בקשה דרך ה-attacker ns נקבל כתובת מזויפת:



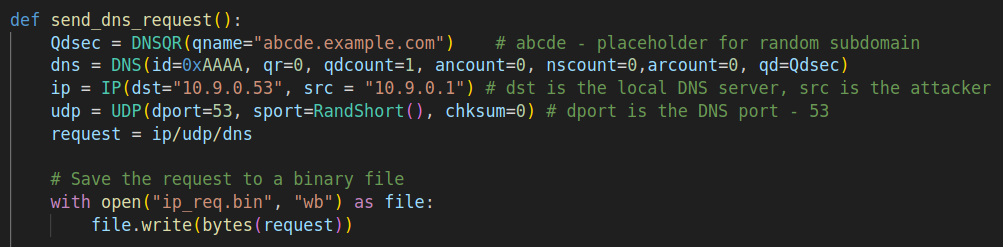
3.2:

לצורך בניית פאקטת ה-DNS, הרצנו את הפקודה ifconfig במכונת ה-attacker כדי למצוא את כתובת ה-IP שלו:



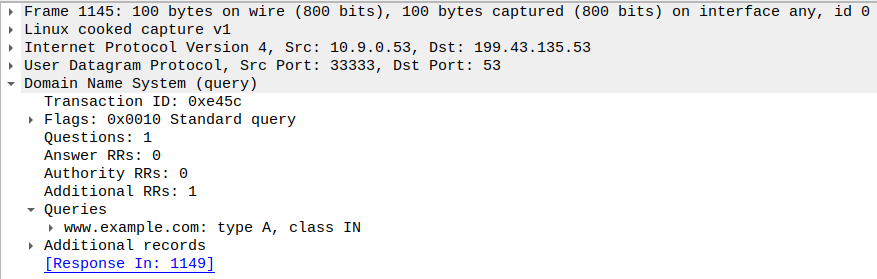
באמצעות scapy ניצור בקשת DNS לכתובת בתצורה XXXXX.example.com.

מטרת הקוד הוא ליצור מעין תבנית לשליחת בקשת DNS מה-Attacker ל-local DNS. כך שבהמשך נשנה את הקידומת של XXXXX.example.com בלבד ונוכל לשלוח בקשות.

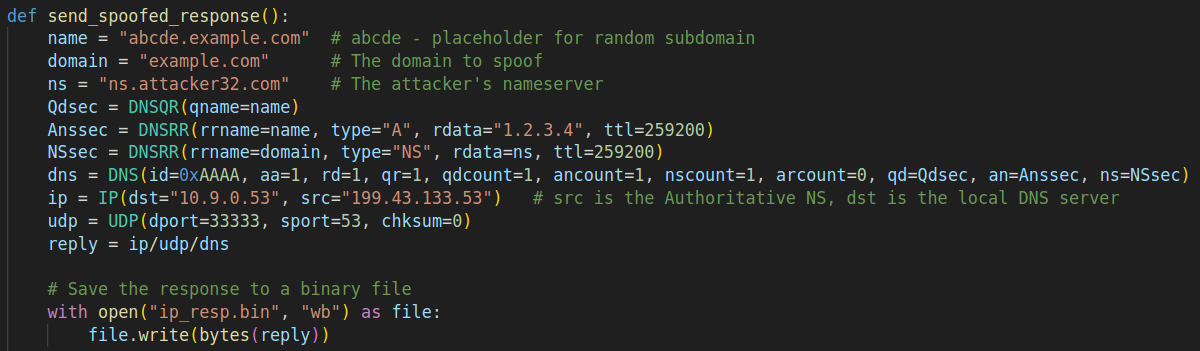


3.3:

בחלק זה נרצה לזייף תשובה לבקשת DNS, כך שהתשובה שלנו תחליף את ה-IP האמיתי של האתר www.example.com, ב-IP של ה-Attacker NS. את התשובה הזו נשלח ל-local DNS במטרה שהוא יכניס את התשובה הזו ל-Cache שלו ויענה לשאילתות הבאות עם ה-IP המזויף שניתן לו.



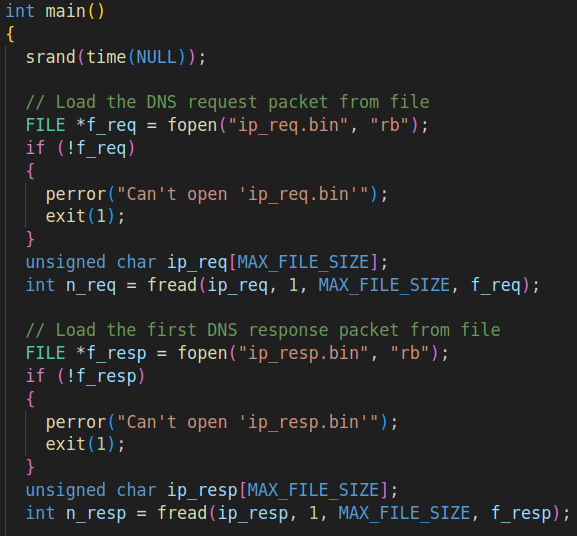
נשים שה-IP של ה-Authoritative Nameserver שה-local DNS פונה אליו כדי לקבל IP Resolution על [www.example.com](http://www.example.com) הוא: 199.43.135.53.



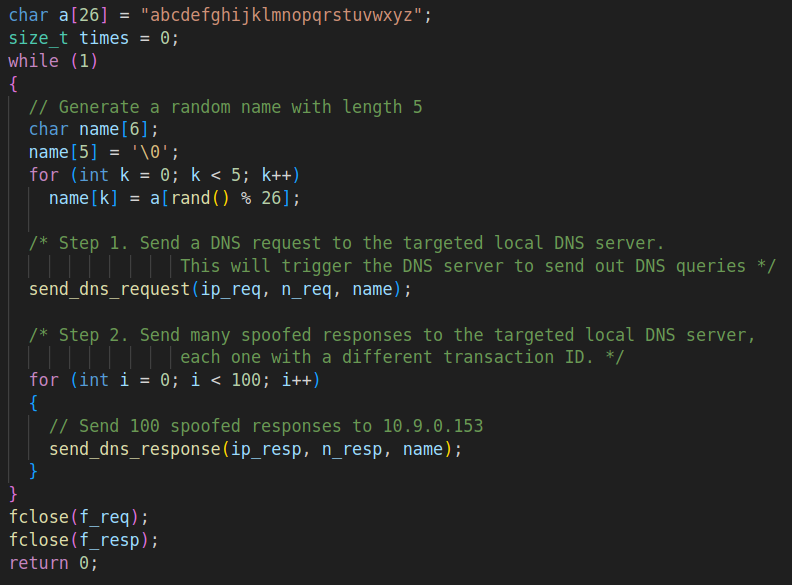
באמצעות הפרטים ממקודם, נבנה תשובת DNS מזויפת, עם פרטי ה-IP המתאימים (כך שהתשובה מגיעה מה-Authoritative NS ויוצאת אל ה-local DNS). והפורטים מתאימים גם כן לבקשה (כמו כן, ה-dport קבוע על 33333 כמו שנטען במטלה).

3.4:

באמצעות הקבצים שיצרנו בעזרת הקוד שכתבנו ב-Python, נבצע את המתקפה:



ראשית, נקרא את תוכן הקבצים שיצרנו ב-Python. הם יהוו בסיס לבקשות ותגובות ה-DNS שנשלח בהמשך.



לאחר מכן, ניצור תחילית בת 5 אותיות רנדומליות ונשלח בקשה לכתובת XXXXX.example.com כאשר XXXXX היא התחילית שיצרנו.

מיד לאחר מכן נשלח 100 תשובות DNS כאשר כל תשובה מכילה transaction id שונה. בתקווה שאחת מהתשובות תכיל את ה-transaction id המתאים.

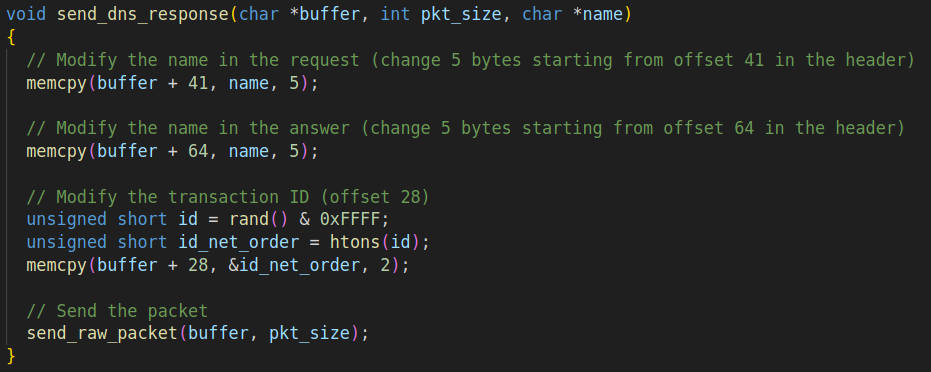
כעת נעבור להסביר את הקוד של השליחת בקשה ושליחת תשובת DNS:

A computer screen with text

Description automatically generated

קטע קוד זה משנה את התחילית של ה-buffer במיקום 41 בייטים מההתחלה לשם אותו שלחנו כארגומנט לפונקציה. ולאחר מכן שולח את הבקשה באמצעות פונקציית send\_raw\_packet.

בעצם הפונקציה משנה את החלק שדיברנו עליו קודם, של XXXXX.example.com ל-“name”.example.com כאשר name זהו צירוף בן 5 אותיות שנבחרו רנדומלית קודם לכן.



ניזכר שתשובת DNS מכילה גם את הבקשה אליה היא עונה.

לכן, נצטרך לשנות את התחילית המתאימה גם במיקום של הבקשה ב-Header וגם במיקום של התשובה ב-Header.

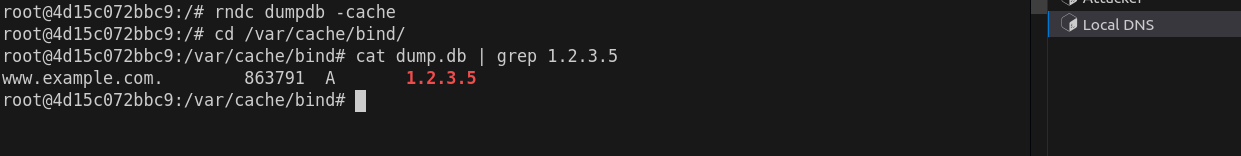
לאחר מכן, נגדיר transaction id רנדומלי – נבצע AND עם 0xFFFF שזהו הערך המקסימלי שיכול להיות ל-transaction id. כך שכל ערך שיחרוג מזה יאופס.

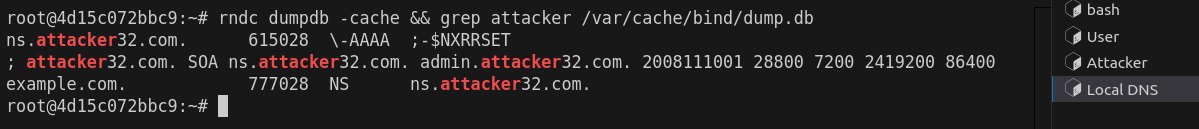
נשנה את ה-transaction id בפאקטה ונשלח אותה באמצעות פונקציית send\_raw\_packet.

כעת נבצע את ההתקפה.

לאחר מס' נסיונות של ה-Attacker לבצע Cache Poisoning, אנחנו יכולים לראות בתמונות הבאות שהוא אכן הצליח.

התחברנו ל-Local DNS וראינו שב-Cache שלו, תחת [www.example.com](http://www.example.com) מופיע ה-IP שהגדרנו.

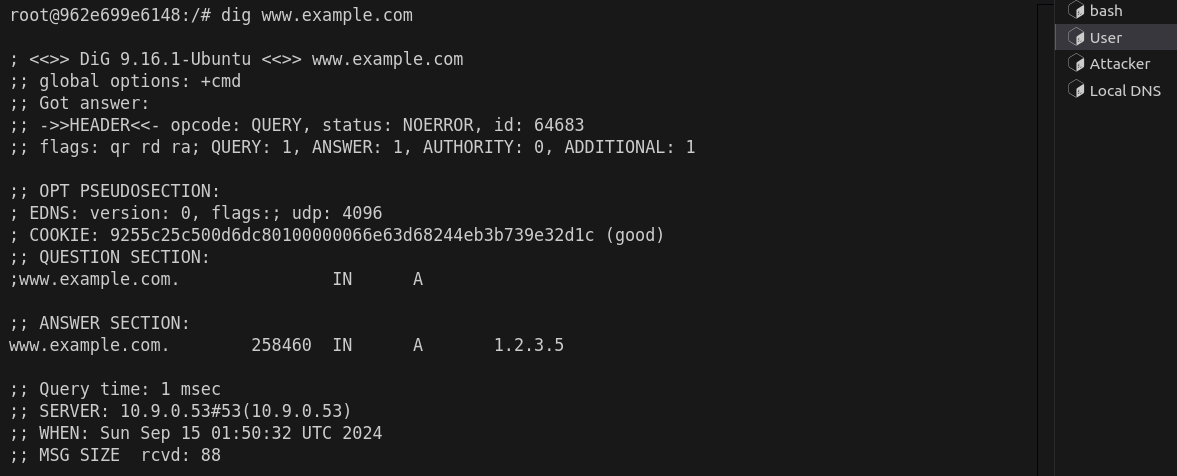




כמו כן, ניתן לראות בשורה האחרונה ש-example.com מופנה ל-ns.attacker32.com. שזו הייתה מטרת כל ההתקפה.

3.5:

כעת נראה דרך ה-User שהמתקפה אכן פעלה:



A screenshot of a computer

Description automatically generated

ב-2 התמונות המצורפות ניתן לראות שאנו מבצעים dig ל-www.example.com באופן רגיל ודרך ה-Attacker NS, ושניהם מחזירים את אותו IP עבור האתר.

משמע שהצלחנו לגרום ל-Local DNS לשמור את הכתובת אותה ה-Attacker NS מחזיר.

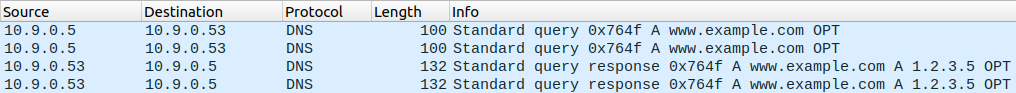
**הקלטות Wireshark:**

A screenshot of a computer

Description automatically generated

לפני שההתקפה הצליחה, ניתן לראות בתמונה דוגמה לכך שה-User מבצע שאילתת DNS על “bvnsi.example.com”, ומיד לאחר מכן נשלחות 100 DNS Responses. כל אחת עם transaction id שונה (מסומן באדום).

לאחר שהצלחנו לבצע Cache Poisoning:



ניתן לראות שכאשר אנו מבצעים שאילתה עבור [www.example.com](http://www.example.com), השאילתה אכן נשלחת ל-10.9.0.53 שזו הכתובת של ה-Local DNS. והוא מחזיר תשובה שהיא כתובת ה-IP שהגדרנו (1.2.3.5).