שיעור 08 – תרגיל מסכם סקאפי

**E.T. COME HOME**

זה היה עוד יום ה' שגרתי במחלקת "חיים תבוניים" של סוכנות החלל האמריקאית, נאס"א, כשלפתע התפרץ למשרד סוכן השטח גרי בצרחות.

"הפעם זה הדבר האמיתי. אני אומר לכם! קיבלנו דיווחים ממטיילים שראו יצורים חשודים במדבר נוואדה. כשהגענו למקום ראינו פיצוץ לבן מטורף. אני יודע שאתם לא תאמינו לי אבל הפעם יש לי הוכחות. משהו נשאר מאחור".

גרי שלף שקית ומתוכה הוציא הארדיסק שבור ומפוחם. הסוכנים החלו לחקור אותו ולנסות לשחזר את המידע שהיה בתוכו.

כעבור שבוע, מעבדת המידע בנאס"א הצליחה לשחזר את כל המידע, ולאחר סינון קפדני נמצאה רק תוכנה אחת שעל פי החוקרים עשויה ללמד אותנו משהו על מה שקרה במדבר נוואדה. על פי ההערכות, הישות שנצפתה ניסתה להשתמש בתוכנה כדי לשוב אל כוכב הלכת ממנו היא הגיעה, אך לא הצליחה.

הפנייה לחניכי מגשימים לא איחרה להגיע. האם תוכלו לעזור לנאס"א לפענח מה עושה התוכנה המסתורית, ויותר חשוב – **האם תוכלו לעזור לישות לחזור לכוכב שלה?**

**איך מתחילים?**

בתיקיית התרגיל בדרייב תוכלו למצוא את התוכנה שחולצה מההארדיסק המסתורי.

**הוראות?**

הוראות זה לחלשים. אתם עוד שניה בוגרי שנה א' של מגשימים! בהצלחה....

**ומה אם נתקעתי?**

אם אתם תקועים יותר מדי זמן ולא יודעים כיצד להתקדם, צרו קשר עם המדריך שלכם לרמז.

**ומה עושים כשמסיימים?**

אם אתם בטוחים שהצלחתם לעזור לישות לחזור לכוכב שלה, אנא הגישו שני דברים – את מסמך זה עם הסבר כיצד פתרתם את התרגיל, ואת קובץ או קבצי הקוד בהם השתמשתם.

**תיאור – כיצד פתרתי את התרגיל:**

| ראשית, משיעורים הבנתי שהIP של כול התוכניות של מגשימים זה 54.71.128.194 אז כאשר כתבתי ip.addr בwireshark ולחצתי על הכפתור באמצע תוכנית קיבלתי את הפלט שלה. מזה הבנתי ברגע הראשון שני דברים, יש לתוכנה שני מודים. אחד הוא "blue" שעובד בפורט 99 ומוד "green" בפורט 98.  אחר כך, קיבלתי הודעה מי אפליקציה של ההודעה שאני צריך לשלוח, ואחר כך אחרי שפירקתי את הקוד עם סקריפט של פייתון:FLY008location\_md5=00000000000000000000000000000000,airport=nevada25.84,time=15:52,lane=earth.jup,vehicle=2554,fly  אחרי הניתוך, אני שלחתי את ההודעה בENCODE בעמצעות SCAPY.  הנה הקוד:from scapy.all import \*  import hashlib  import socket    ALIEN\_IP = "54.71.128.194"  ALIEN\_UDP\_PORT = 99  """  function that prints source and dest ip of a packet  params: packet  returns: -  """  def print\_packet(packet):  print(packet[IP].src, "-", packet[IP].dst)  """  function that decodes the encoded message from alienClient  params: text (string) encoded message from alien client  returns: the decoded message  """    def decode\_message(text):  allowed\_letters = (' ', ',', '!', '.', ':', '\*', ';', '/', '=')  key = int(text[3:6])  result = ''  for index, letter in enumerate(text[6:]):  if index % 2 == 1 or letter in allowed\_letters or letter.isnumeric():  result += letter  else:  result += chr((ord(letter) - 97 - key) % 26 + 97)  print(result)  print("=====================")  return result  """  function that makes sniffing stop when there are 10/10 location data packets  params: packets  returns: true if there are 10 location data packets, else, false  """  def stop(packet):  location\_data\_bool = False  try:  location\_data\_bool = "location data 10/10:" in decode\_message(packet[Raw].load.decode())  except Exception:  pass  return location\_data\_bool    """  function that filters packets from alien client  params: packet  returns: true if packet is from alien client, else, false  """  def alien\_filter(packet):  try:  return packet[IP].src == ALIEN\_IP and packet[UDP].sport == ALIEN\_UDP\_PORT  except IndexError:  return False    """  function that encodes the message to be sent to alien client  params: text (string) message to be encoded  returns: the encoded message  """  def encode\_message(text):  allowed\_letters = (' ', ',', '!', '.', ':', '\*', ';', '/', '=', "b'", "'", "\_")  result = ''  key = 8  for index, letter in enumerate(text):  if index % 2 == 1 or letter in allowed\_letters or letter.isnumeric():  result += letter  else:  result += chr((ord(letter) - 97 + key) % 26 + 97)  return result    """  function that sends the final packet to alien client  params: final\_packet (string) the final packet to be sent, \_sport (int) source port of the packet  returns: -  """  def send\_final(final\_packet, \_sport):  print("Sending final packet:", final\_packet)  full\_msg = Ether() / IP(*dst*=ALIEN\_IP) / UDP(*dport*=ALIEN\_UDP\_PORT, *sport*=\_sport) / Raw(*load*=final\_packet)  srp1(full\_msg, *verbose*=0, *timeout*=2)    """  Main function  """  def main():  big\_param\_str = ""  packets = sniff(*lfilter*=alien\_filter, *prn*=print\_packet, *stop\_filter*=stop)    for packet in packets:  my\_port = packet[UDP].dport  result = decode\_message(packet[Raw].load.decode())  if "location data" in result:  big\_param\_str += result[-10:]    md5\_hash = hashlib.md5(big\_param\_str.encode()).hexdigest()    raw\_message = "location\_md5"+md5\_hash+"airport=nevada25.84,time=15:52,lane=earth.jup,vehicle=2554,fly"  final\_packet = "FLY008" + encode\_message(raw\_message)    send\_final(final\_packet, my\_port)      if \_\_name\_\_ == '\_\_main\_\_':  main()    (עשיתי תיעוד🥳🥳🥳) |
| --- |
| *תשובה* |

צרפו צילום מסך של סיום התרגיל:

