



S3/L5

ATTACCO DOS

# TRACCIA

Gli attacchi di tipo Dos, ovvero denial of services, mirano a saturare le richieste di determinati servizi rendendoli così indisponibili con conseguenti impatti sul business delle aziende.  
L'esercizio di oggi è scrivere un programma in Python che simuli un UDP flood, ovvero l'invio massivo di richieste UDP verso una macchina target che è in ascolto su una porta UDP casuale.

## Requisiti:

- Il programma deve richiedere l'inserimento dell'IP target.
- Il programma deve richiedere l'inserimento della porta target.
- La grandezza dei pacchetti da inviare è di 1 KB per pacchetti
- Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare.

01

Questo codice Python implementa un attacco di tipo UDP flood attraverso l'uso di socket e thread multipli. Innanzitutto, il programma importa le librerie necessarie come **socket** per la comunicazione di rete, random per la generazione di dati casuali, e **threading** per gestire più operazioni simultaneamente. È utilizzato anche **colorama** per colorare l'output per migliorarne l'accessibilità.

02

Nella funzione **udp\_flood(target ip, target port, num packets)**, viene creato un **socket UDP (socket.SOCK\_DGRAM)** per inviare pacchetti di dati. Viene generato un byte array di 1KB di dati casuali che verranno inviati ripetutamente al target specificato attraverso il metodo `sendto`. Se l'invio ha successo per il numero desiderato di pacchetti, viene stampato un messaggio verde di completamento dell'attacco; altrimenti, viene gestito qualsiasi eccezione mostrando un messaggio di errore in rosso.

03

Nella funzione **main()**, l'utente inserisce l'indirizzo IP e la porta del target, oltre al numero di pacchetti da inviare e il numero di thread da utilizzare per l'attacco. Se i valori inseriti sono validi, il programma divide equamente il numero di pacchetti tra i thread creati, ognuno dei quali esegue la funzione **udp\_flood** in parallelo. Al termine dell'attacco, il programma aspetta che tutti i thread completino prima di terminare.



CODE

# THREADING

Utilizzare il threading nel contesto di uno script di attacco di tipo UDP flood significa creare più thread per inviare pacchetti contemporaneamente (allo stesso tempo) piuttosto che in sequenza (uno dopo l'altro).

Questo simula “un'inondazione” di pacchetti più intensa verso il bersaglio, rendendo l'attacco potenzialmente più efficace aumentando il volume e la velocità dei pacchetti inviati.

Vantaggi dell'uso del threading in un attacco UDP flood:

- **Invio parallelo:** Più thread funzionano in parallelo, ciascuno inviando pacchetti contemporaneamente.
- **Aumento dell'invio:** Un maggior numero di pacchetti può essere inviato in un periodo di tempo più breve perché più thread lavorano simultaneamente
- **Attacco più efficace:** Aumentando il numero di pacchetti inviati per unità di tempo, l'attacco può potenzialmente sopraffare il bersaglio in modo più efficace.

Utilizzando questa tecnica, l'attacco riesce a sfruttare meglio le risorse disponibili e a generare un traffico più elevato verso il bersaglio, aumentando così le probabilità di successo nell'intasare e disturbare il servizio.

# ESECUZIONE CODICE

```
(kali㉿kali)-[~/Desktop]
$ python S3L5.py
Inserisci l'IP target: 127.0.0.1
Inserisci la porta target: 1234
Inserisci il numero di pacchetti da inviare: 10
Inserisci il numero di threads con cui inviarli: 2
Attacco UDP flood completato con successo.
Attacco UDP flood completato con successo.
```

## UDP FLOOD COMPLETATO

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	127.0.0.1	127.0.0.1	UDP	1066	40611 → 1234 Len=1024
2	0.000019066	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (P)
3	0.000142884	127.0.0.1	127.0.0.1	UDP	1066	40611 → 1234 Len=1024
4	0.000146735	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (P)
5	0.000155056	127.0.0.1	127.0.0.1	UDP	1066	40611 → 1234 Len=1024
6	0.000157700	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (P)
7	0.000161107	127.0.0.1	127.0.0.1	UDP	1066	40611 → 1234 Len=1024
8	0.000164451	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (P)
9	0.000167952	127.0.0.1	127.0.0.1	UDP	1066	40611 → 1234 Len=1024
10	0.000171183	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (P)
11	0.000182787	127.0.0.1	127.0.0.1	UDP	1066	57856 → 1234 Len=1024
12	0.000827543	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (P)
13	0.000842225	127.0.0.1	127.0.0.1	UDP	1066	57856 → 1234 Len=1024
14	0.000846243	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (P)
15	0.000850314	127.0.0.1	127.0.0.1	UDP	1066	57856 → 1234 Len=1024
16	0.000853573	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (P)
17	0.000857304	127.0.0.1	127.0.0.1	UDP	1066	57856 → 1234 Len=1024
18	0.000859877	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (P)
19	0.001007203	127.0.0.1	127.0.0.1	UDP	1066	57856 → 1234 Len=1024
20	0.001010874	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (P)

```
(kali㉿kali)-[~/Desktop]
$ python S3L5.py
Inserisci l'IP target: 127.0.0.1
Inserisci la porta target: 1234
Inserisci il numero di pacchetti da inviare: 10
Inserisci il numero di threads con cui inviarli: 2.
Errore input: invalid literal for int() with base 10: '2.'
```

## ERRORE IN INPUT



# OUR VISION

**01**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint

**02**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint



# OUR MISSION

**01**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco

**02**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco

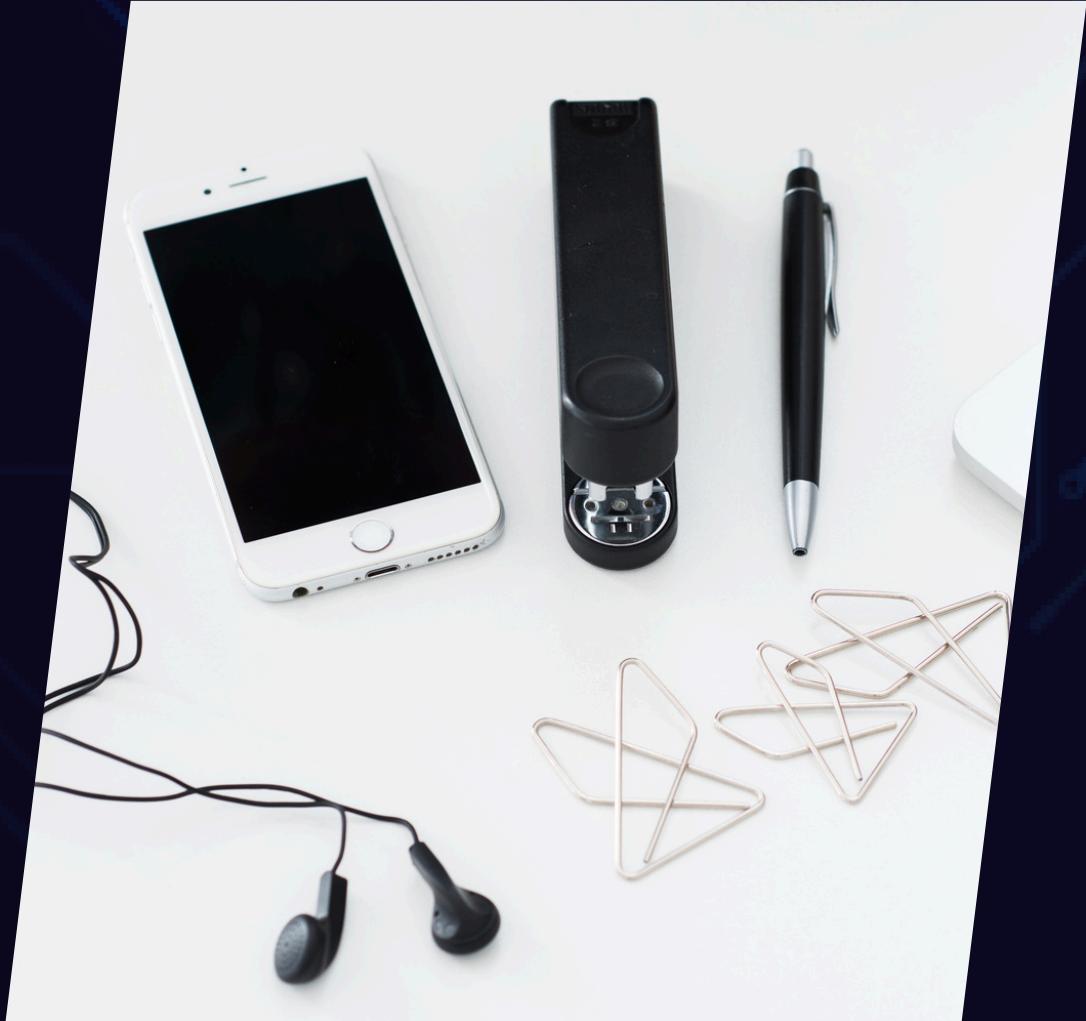
**03**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco





# PORTFOLIO



*Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit*



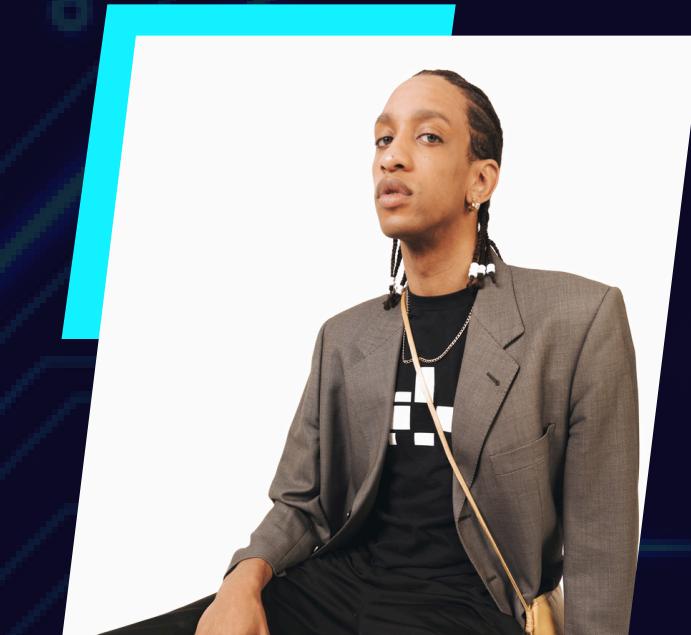
# MEET OUR TEAM

**Alfredo Torres**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim

**Bailey Dupont**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim

**Daniel Gallego**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim

**Adeline Palmerston**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim



# CONTACT US



+123-456-7890



[www.reallygreatsite.com](http://www.reallygreatsite.com)



[hello@reallygreatsite.com](mailto:hello@reallygreatsite.com)



123 Anywhere ST., Any City, ST 12345



# THANK YOU