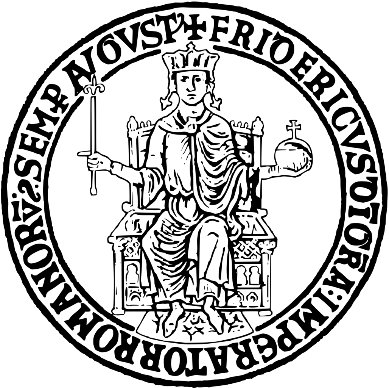
Hackerman

Reti Di Calcolatori

Professore: Alessio Botta



Gianmarco Lembo - N86003940

Federico Gargiulo - N86002884

Cristian Carrella - N86003677

1. DESCRIZIONE DELLA BOTNET

Questo progetto si divide in due eseguibili scritti in python: Bot e BotMaster.

Il **botmaster** può mandare istruzioni al bot che esegue comandi di sistema a seconda del sistema operativo su cui sta girando, il tutto viene gestito tramite linea di comando personalizzata.

I comandi che possiamo eseguire sono i seguenti:

1. infoOs - Ottiene informazioni sul sistema operativo e sull hardware.
2. search [string] - fa una ricerca ricorsiva di string dal path corrente.
3. rf - ottiene i file ordinati per ultima modifica - vecchi 🡪 nuovi (da una certa data o, di default, dal 1990).
4. get [fileName] - Scarica un file presente sulla macchina dove gira il bot.
5. cd - Usato per navigare nelle directory.
6. dir/ls - Restituisce i file e le cartelle trovate nella cartella attuale.
7. pwd - Restituisce la directory attuale.
8. cls - usato per pulire la console.
9. esc - chiude il Botmaster ma non il bot e lo rimette in attesa.
10. help
11. nsf [cmd] - manda un comando eseguito direttamente sulla shell.
12. next - chiude la connessione con il bot corrente e stabilisce la connessione con il successivo in lista.

**CONNESSIONE**

Il **Bot** appena si avvia tenta una connessione al server e fin quando non la ottiene continua a provare con intervalli regolari.

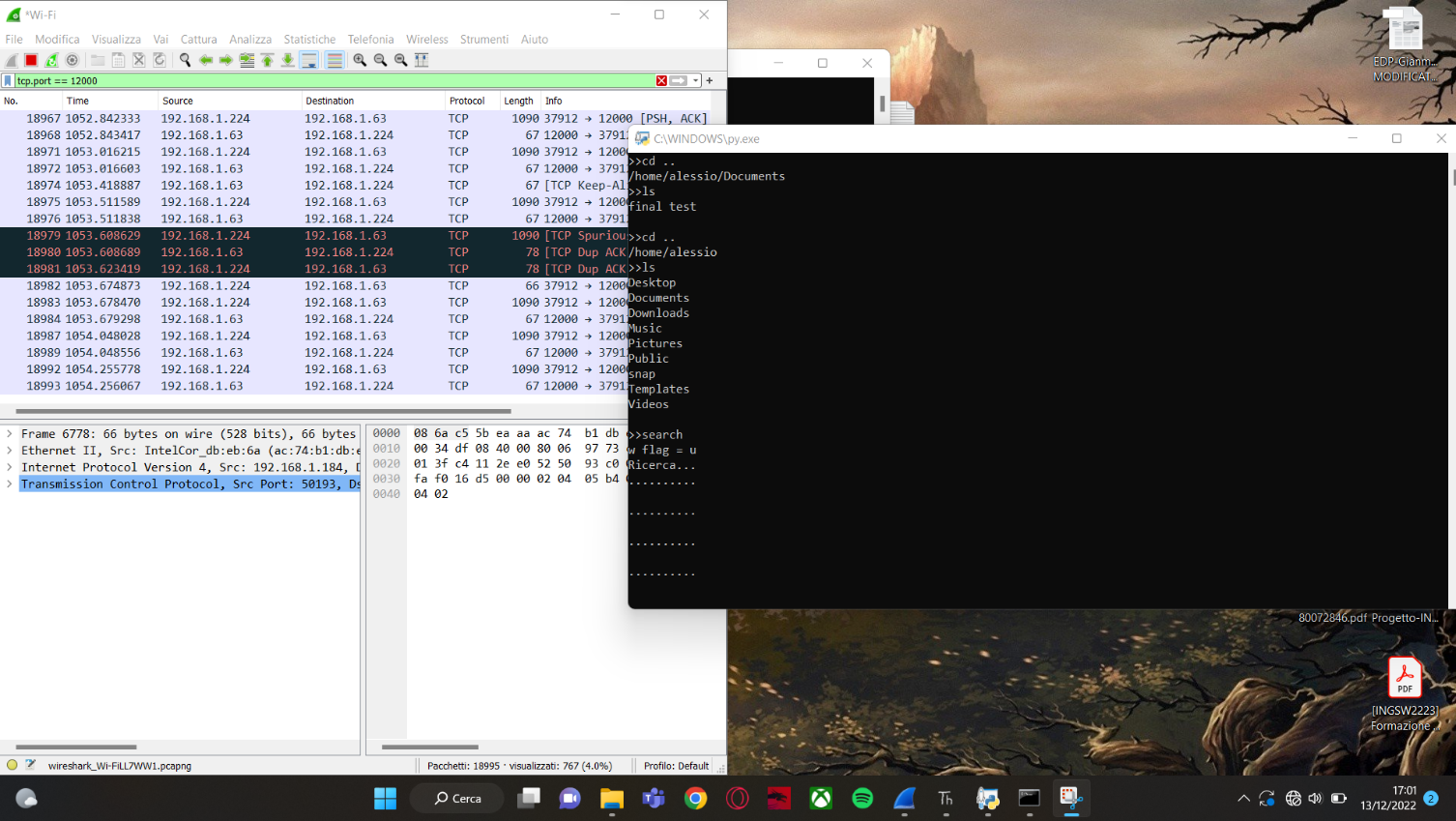
Quando ottiene la connessione invia dati per sapere su che sistema ci troviamo (Unix - Windows) e si mette in attesa.

Il **Botmaster** invece apre una socket e la mette in ascolto in modo da avere una lista di bot e collegarci al successivo tramite comando next che chiude la connessione col bot attuale e si collega alla successiva.

**ERROR RECOVERY**

Sia il bot che il botmaster hanno un sistema di error recovery molto rudimentale ma funzionante (tranne per un caso descritto in seguito) in pratica chiudono la socket e riprovano la prima connessione.

2. REPORT DELLA PROVA

Abbiamo configurato la macchina con l'IP statico fornitoci dal professore, nonostante ciò per i primi 40 minuti il computer non era visibile dalla rete anche attraverso comandi di ping eseguito su macchine diverse connesse alla stessa rete, a seguito di ciò abbiamo deciso di cambiare computer su cui far girare il "botMaster.py" configurando nuovamente (e nello stesso modo) gli ip statici.

A questo punto, avviando il bot master, grazie all'implementazione del sistema di connessione del bot - descritta precedentemente - riusciamo a stabilire la connessione con il client. Dopo qualche comando di prova decidiamo di lanciare:

- qualche comando di navigazione nel FS (e.g. cd, ls, nsf ls -l -a...)

- un comando "search" sulla directory "/home/alessio" per avere un'idea veloce dei file presenti a partire dalla home;

- un comando "infoOs" per recuperare i dati relativi alla macchina e al OS da cui abbiamo ricavato il seguente output:

################################################################################

Architecture: 64bit

Macchine: x86\_64

System name: Linux

Operating system release: 5.15.0-52-generic

Operating system version: #58-Ubuntu SMP Thu Oct 13 08:03:55 UTC 2022

Node: Ubuntu-20-04-LTS

Platform: Linux-5.15.0-52-generic-x86\_64-with-glibc2.35

Processor: x86\_64

Info CPU

Architecture: x86\_64

CPU op-mode(s): 32-bit, 64-bit

Address sizes: 39 bits physical, 48 bits virtual

Byte Order: Little Endian

CPU(s): 4

On-line CPU(s) list: 0-3

Vendor ID: GenuineIntel

Model name: Intel(R) Core(TM) i7-8569U CPU @ 2.80GHz

CPU family: 6

Model: 142

Thread(s) per core: 1

Core(s) per socket: 4

Socket(s): 1

Stepping: 10

BogoMIPS: 5638.69

Hypervisor vendor: KVM

Virtualization type: full

L1d cache: 128 KiB (4 instances)

L1i cache: 128 KiB (4 instances)

L2 cache: 1 MiB (4 instances)

L3 cache: 32 MiB (4 instances)

NUMA node(s): 1

NUMA node0 CPU(s): 0-3

…

Info mem

total used free shared buff/cache available

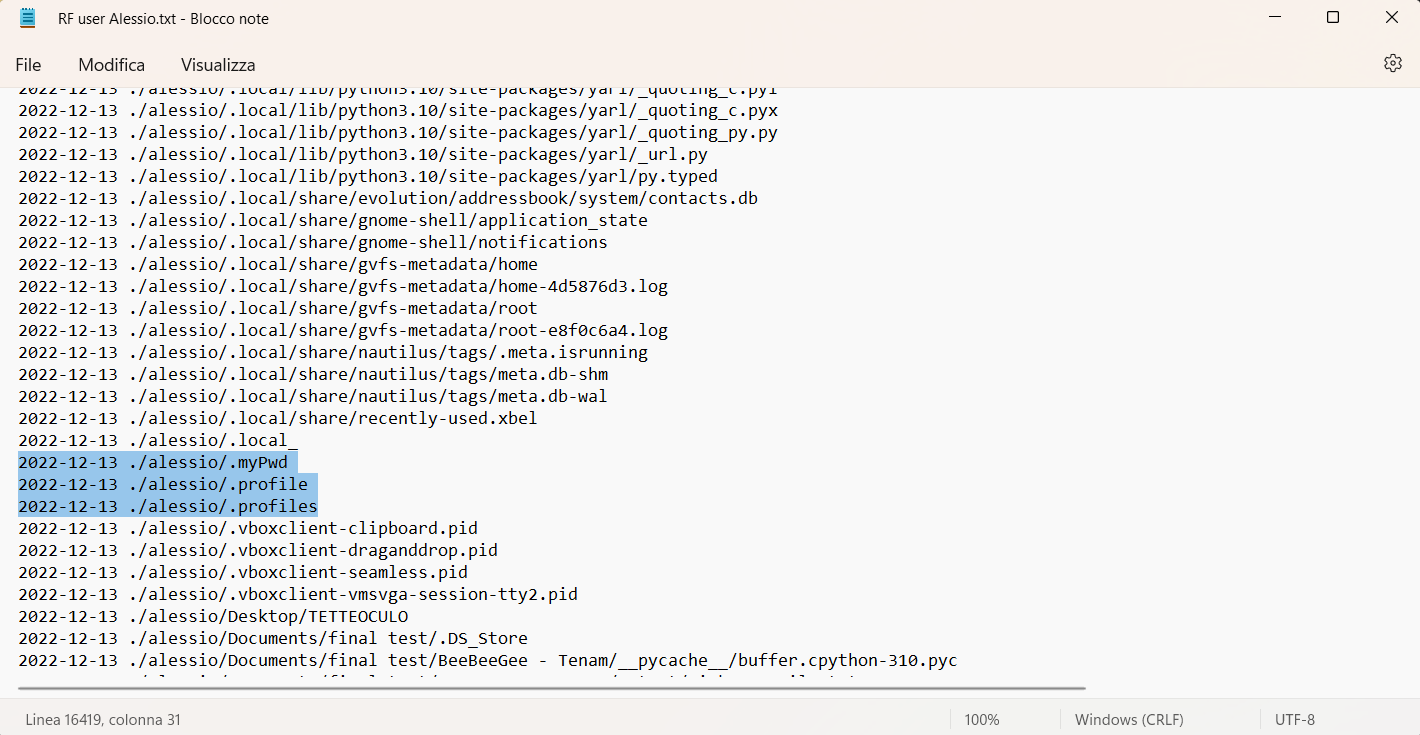
Mem: 3923 1122 1753 17 1047 2435

Swap: 2679 367 2312

################################################################################

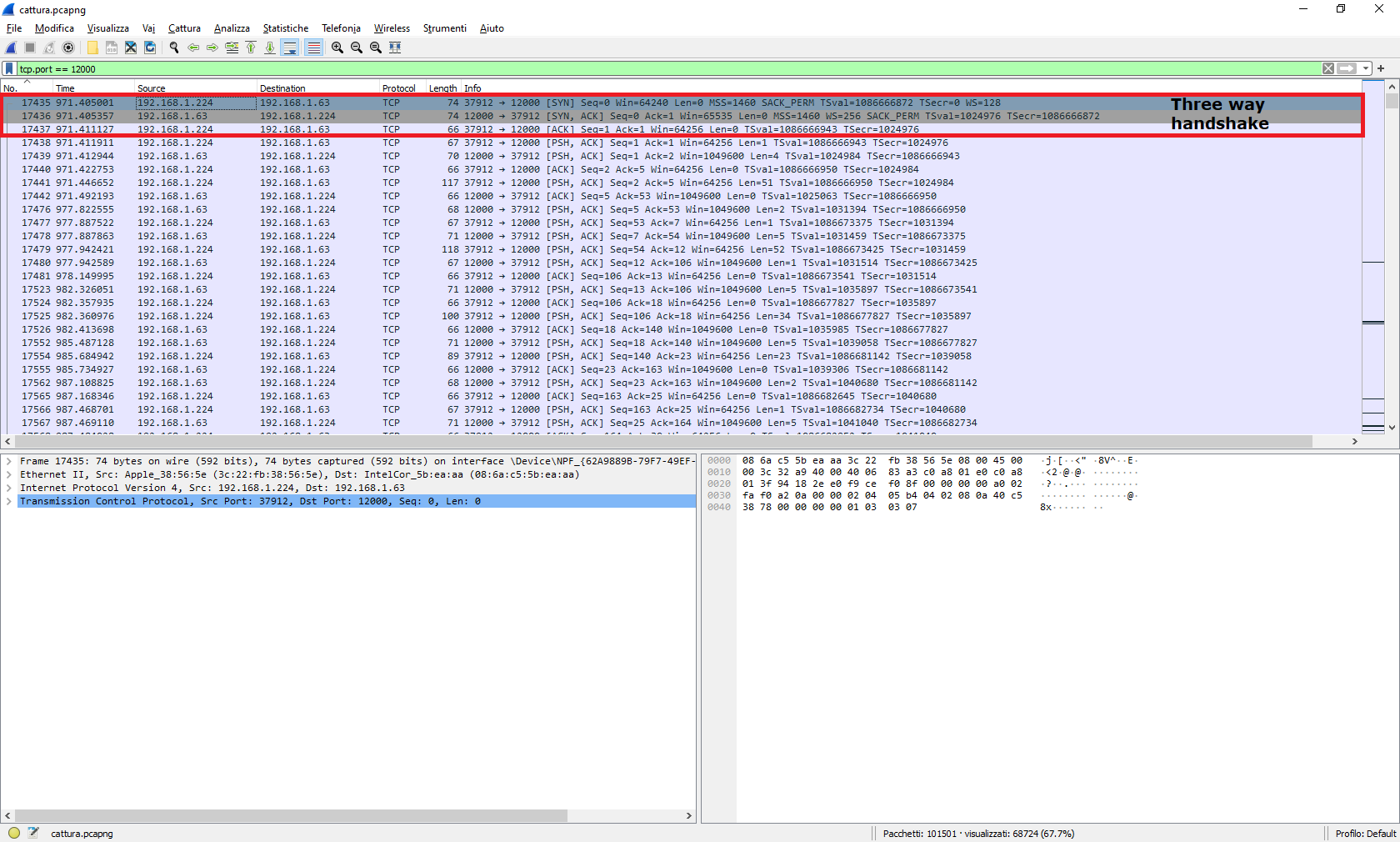
Durante la ricerca dei file nascosti la rete è crashata, ciò ha compromesso il funzionamento del bot sulla macchina "192.168.1.188". Evidentemente il bot è rimasto in attesa su una recv(). L'errore fatale è stato chiudere il botMaster, poichè se non fosse stato terminato la connessione si sarebbe ristabilita al ripristino dall'access point senza causare malfunzionamenti (così come emerso dalle prove effettuate in seguito).

Immagine che contiene testo

Descrizione generata automaticamenteAll'avvio della macchina "192.168.1.224" siamo riusciti a connetterci senza problemi. Abbiamo provato ad usare la strategia della ricerca dei file più recenti utilizzando il comando "rf 2022-12-13" sulla directory "/home/alessio/" da questo abbiamo individuato alcuni files interessanti come: ".myPwd" (contenente "SBAGLIATO! VEDI ALTROVE"), ".profile".

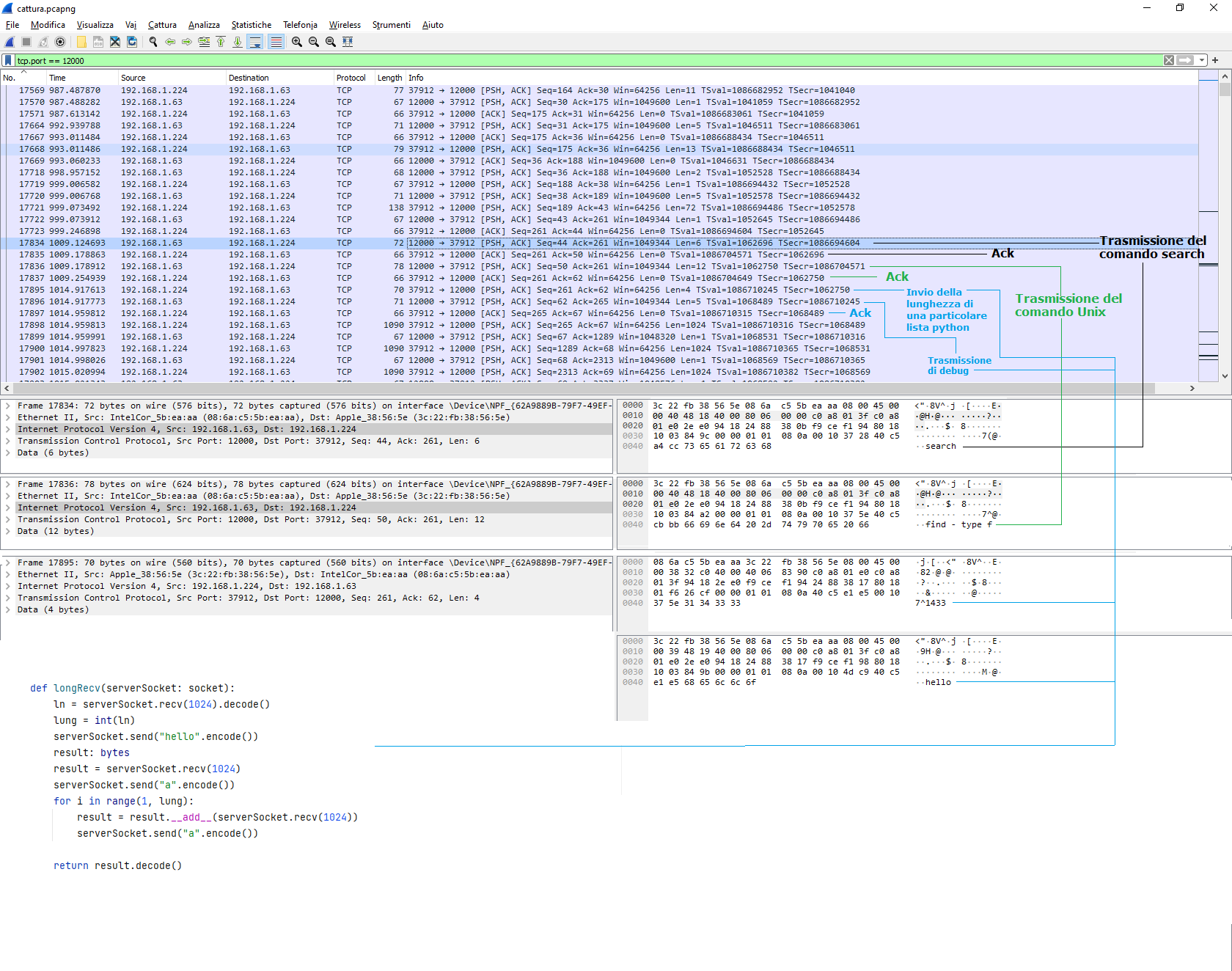
Analizzando le specifiche raccolte abbiamo notato che i due computer sui quali giravano i bot erano estremamente simili infatti le uniche informazioni che differivano erano i dati di memoria libera/allocata.

3. ANALISI DEI PACCHETTI

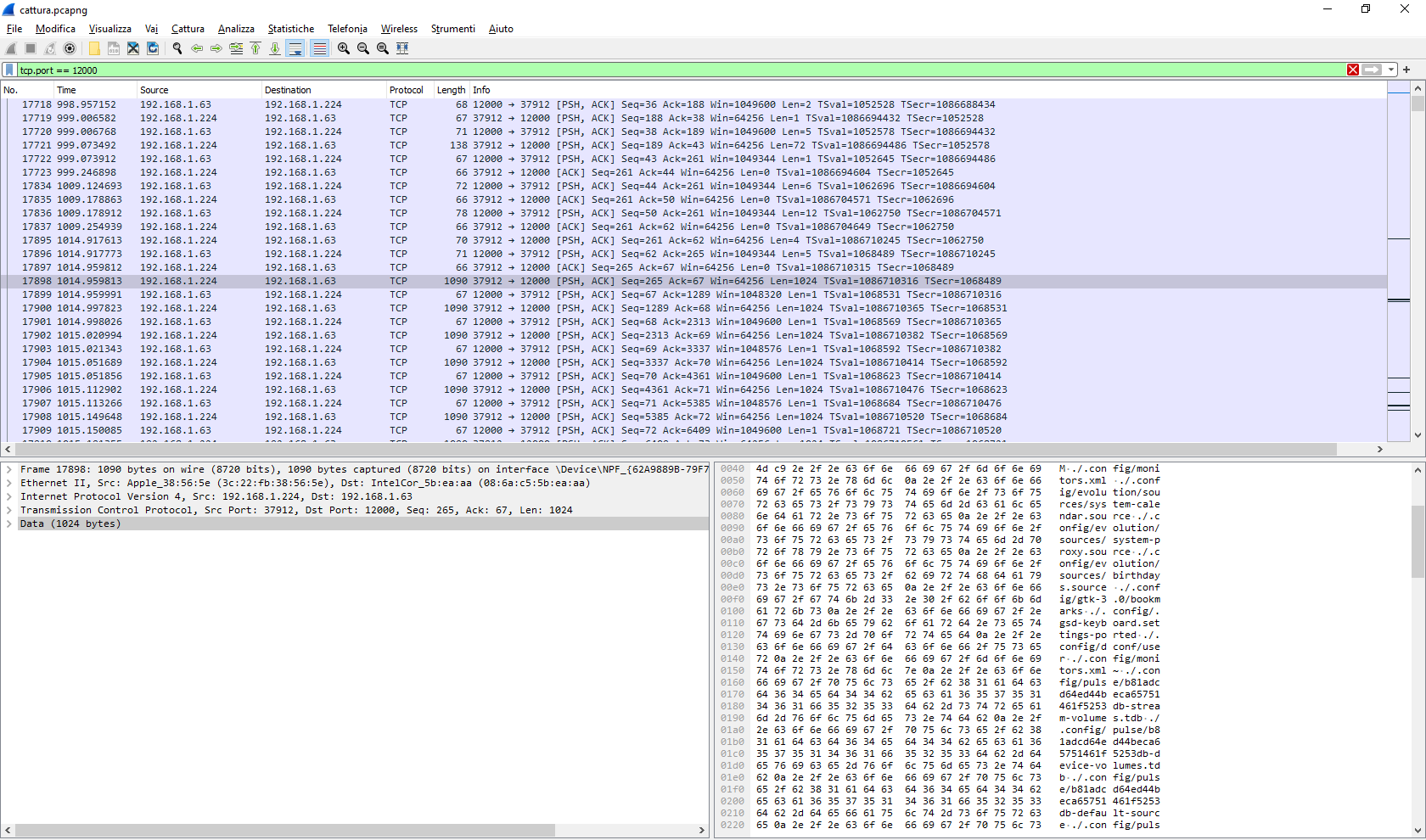
Durante quasi tutta la durata della prova abbiamo tenuto aperto wireshark per la cattura dei pacchetti. Dall'analisi effettuata in seguito abbiamo individuato il three way handshake effettuato tra la bot e il botmaster al momento della connessione iniziale.

Inoltre abbiamo selezionato ed individuato l'invio di una "search"

come comando campione da analizzare.



Dal frame 17898 vengono in successione inviati e ricevuti pacchetti riguardanti il comando search precedentemente lanciato.



Vedendo nel dettaglio un singolo pacchetto possiamo notare i campi compilati del livello rete e livello trasporto.

Nel layer di rete possiamo notare: l’indirizzo sorgente 192.168.1.224 e l’indirizzo destinazione 192.168.1.63, la versione (IPV4), il TTL (64), il protocollo soprastante (TCP), il checksum.

Nel layer di trasporto possiamo notare: Porto sorgente 37912 e porto di destinazione 12000, numero di seq relativo (265) / reale (4191089048), ack (67), len (1024), TCP segment length (1024), window (502), checksum, urgent pointer (0), TCP payload (1024).

Immagine che contiene testo

Descrizione generata automaticamente