SAPIENZA
UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INFORMATICA

Botnet

SICUREZZA

Professor:
Emiliano Casalicchio

Student:
Edoardo Allegrini

Anno Accademico 2022/2023

Contents

1 Che cosa è una botnet ? (piccolo inciso) **2**

2 Command & Control **2**

3 Bot **2**

4 Attività **2**

5 Development **3**

5.1 Command & Control 3

5.1.1 handle_bot_connection 3

5.1.2 cc_cli 3

5.2 Bot 3

5.2.1 servers 3

6 Conclusioni ed esempio **4**

1 Che cosa è una botnet ? (piccolo inciso)

Le botnet si verificano comunemente negli attacchi informatici di oggi, con conseguenti gravi minacce alle risorse della rete e alle proprietà dell'organizzazione. Con il termine botnet ci si riferisce a raccolte di computer compromessi (Bots) che sono controllati da remoto da un'infrastruttura comune di comando e controllo (C&C). Il C&C viene utilizzato per impartire comandi ai bot i quali eseguono attività dannose come gli attacchi DDoS (Distributed Denial of Service), l'invio di grandi quantità di SPAM e altro.

2 Command & Control

L'architettura del C&C che ho implementato prevede l'esecuzione di uno script python, "main.py", il quale gestisce contemporaneamente le connessioni da parte dei bot e l'invio di comandi da far eseguire ai bot. Ho previsto la coesione delle due attività suddette in modo tale che, con l'ausilio di Threads, mentre il C&C ha una connessione attiva con un bot e sta comunicando con esso, α possa in concomitanza registrare la connessione di un altro bot β alla botnet.

3 Bot

L'architettura della entità Bot che ho implementato prevede l'esecuzione di uno script python, "main.py", il quale appena eseguito inizializza i server ftp, http e irc sulle rispettive porte 21, 80, 6667 ognuno gestito da un Thread diverso in modo tale da mantenerli attivi e funzionanti contemporaneamente. Dopo aver aperto le tre porte si connette al C&C per informarlo che è attivo e in ascolto sui tre servizi.

4 Attività

Di seguito la lista delle azioni che i bot possono effettuare su richiesta del C&C:

- 'b', invia un batch di email
- 'c', invia un comando personalizzato (DDoS)
- 'i', ottieni informazioni sulla configurazione software e hardware del sistema
- 'idle', interrompi qualunque azione il bot stia svolgendo
- 'kill', uccidi il bot
- 'exit', esci dalla CLI con il bot corrente

5 Development

5.1 Command & Control

5.1.1 handle_bot_connection

La funzione handle_bot_connection() instanzia una socket con cui accetta nuove connessioni, ricevuta una connessione memorizza le info relative all'ip del bot e alle porte che ha aperto da lui stesso comunicate, per ogni porta aperta dal bot inizializza uno stato ad 'idle' che verrà successivamente aggiornato in base all'azione che sta svolgendo il bot.

5.1.2 cc_cli

La funzione cc_cli() gestice la Command Line Interface (CLI) del C&C. La lista dei comandi che ho previsto per il C&C sono:

1. "irc", connessione con un bot sulla porta 6667 utilizzando il protocollo IRC, irc_client_program()
2. "http", connessione con un bot sulla porta 80 utilizzando il protocollo http, http_client_program()
3. "ftp", connessione con un bot sulla porta 21 utilizzando il protocollo ftp, ftp_client_program()
4. "dump", printa le informazioni (ip, ports, status) su tutti i bot appartenenti alla botnet
5. "exit", termina l'esecuzione del C&C

irc_client_program && ftp_client_program Effettua la connessione, mediante una socket, con il bot α specificato, se l'operazione genera un errore dato dal fatto che il bot non è più operativo, elimina α dal database della botnet; altrimenti invia i comandi desiderati al bot α, ritorna la risposta (se presente) e aggiorna l'attività corrente di α

http_client_program Comunica con il bot mediante richieste http al server instaurato sulla porta 80 da http_server Se l'azione da far performare al bot α è 'i' invia una richiesta http GET /hwsw e printa il risultato; altrimenti per tutti gli altri comandi esegue una richiesta POST specificando nel body la tipologia.

5.2 Bot

Il bot, dopo aver attivato i servizi ftp, http e irc ed aver comunicato al C&C le sue informarzioni, attende ordini da eseguire. I tre servizi sono gestiti da tre script ausiliari che ho implementato contenuti nei file ftp_server, http_server ed irc_server.

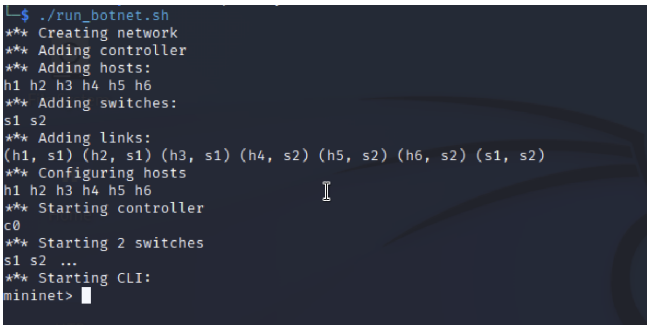
5.2.1 servers

I server handler di ftp e irc sono simili, essi gestiscono la comunicazione con il C&C mediante delle socket, per quanto concerne il server http invece ho deciso di implementare una classe apposita da me denominata "my_HTTPServer_RequestHandler" la quale opera come un classico http server accettando richieste GET e POST. 5.1.2

6 Conclusioni ed esempio

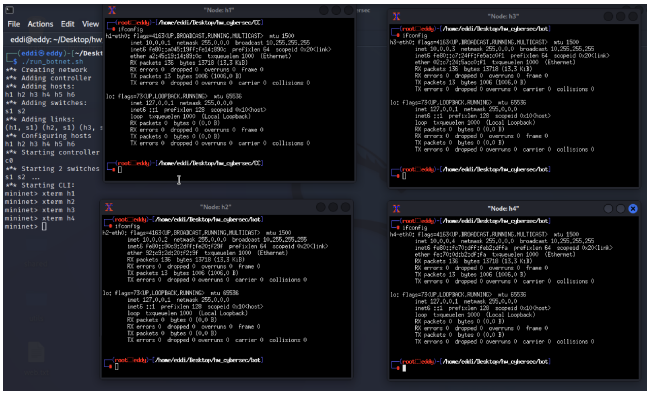
Per simulare una rete quasi indistinguibile da una botnet ho utilizzato una libreria python chiamata mininet con la quale ho creato un treatro formato da sei hosts, per semplicità ho deciso di scegliere un numero basso quale il sei ma ovviamente è solo uno scenario di testing. Eseguendo los cript run_botnet.sh viene instanziata la rete seguendo la topologia sopra indicata.

Figure 1: Esecuzione di run_botnet.sh



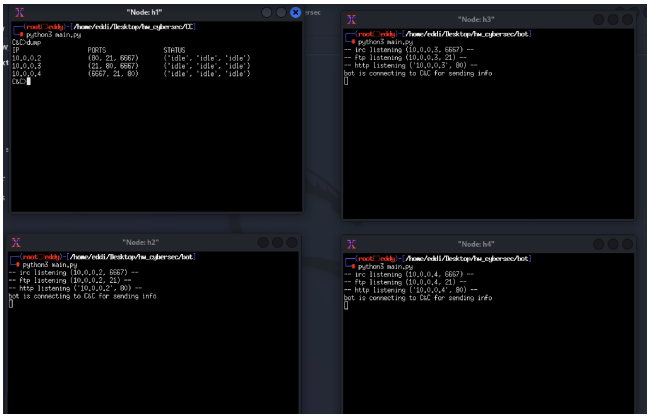
Nell'esempio di funzionamento utilizzerò quattro hosts h1 (C&C), h2 (bot), h3 (bot), h4 (bot) con rispettivi ip 10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.4.

Figure 2: Topologia



Come si può notare in 3 i tre bot sono attivi ed hanno inviato le proprie info al C&C il quale le ha memorizzate

Figure 3: Connessione dei bot al C&C



Di seguito 4 riporto il C&C che invia ad h2, con ip 10.0.0.2, una richiesta mediante irc per performare un attacco DDoS verso google.com

Figure 4: Attacco DDoS