SAPIENZA
UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INFORMATICA

Homework: Botnet

SICUREZZA

Professore:
Emiliano Casalicchio

Studente:
Antonio Pietro
Romito - 1932500

Anno Accademico 2022/2023

1 Introduzione

La traccia scelta è la numero uno, la quale prevede la progettazione di una botnet. Essa è stata scritta in python ed è articolata in due componenti principali: il centro di comando e controllo (C&C), il quale si occupa di inviare istruzioni ai vari bot connessi attraverso un'interfaccia testuale, e i bot, ossia i veri e propri agenti di attacco.

Le componenti comunicano tra loro in due modi:

- Attraverso dei socket stream per l'aggiunta e la rimozione dei bot dalla botnet.
- Attraverso delle richieste HTTP effettuate dal C&C per inviare comandi ai vari bot.

2 Implementazione

Il codice sorgente si trova dentro la directory 'hw/'.

2.1 Centro di comando e controllo (CnC.py)

L'esecuzione del C&C avviene attraverso due thread paralleli: uno che gestisce l'aggiunta e l'eliminazione dei bot dalla botnet ed uno che permette all'utente di gestire i bot attraverso una CLI.

Connessione/disonnessione bot Questo thread rimane in background fin tanto che il C&C è in esecuzione. Il suo scopo è tenere aggiornato il dizionario dei bot connessi alla botnet. Per raggiungere tale fine viene utilizzata una semplice socket in ascolto su una porta nota ai bot. Ogni volta che un bot si connette a questa socket verifica la presenza del suo indirizzo IP all'interno del dizionario: se è già presente lo rimuove (disconnessione), altrimenti lo aggiunge (connessione) usando l'indirizzo IP come chiave e la porta su cui il bot è in ascolto come valore. Viene utilizzato un dizionario perché si presume che su una macchina ci possa essere solo bot in esecuzione in un determinato istante, così facendo l'accesso ai dati relativi al bot è diretto e non sequenziale.

Inoltre alla chiusura del programma, se ci sono bot connessi, il dizionario dei bot viene scritto su un file JSON. All'avvio del C&C invece viene letto il file e per ognuno dei bot presenti in esso viene inviata una richiesta HTTP al path '/status': se viene ricevuta una risposta il bot viene aggiunto al dizionario dei bot attivi altrimenti viene scartato. In questa maniera è possibile memorizzare i bot che sono attivi quando il programma viene chiuso e non è necessario che essi si riconnettano al C&C attivamente.

CLI Questo thread a differenza del primo viene eseguito in foreground e consente all'utente di effettuare vari comandi per controllare i bot connessi. Essi sono:

- **commands**: elenca tutti i comandi disponibili per gestire la botnet.
- **help**: stampa la descrizione di un comando specifico.
- **bots**: elenca tutti i bot connessi alla botnet specificandone l'indirizzo IP e porta su cui sono in ascolto.
- **info**: invia una richiesta GET ad uno specifico bot (dato il suo IP) per ottenere informazioni riguardo la sua configurazione hardware e software.
- **status**: invia una richiesta GET ad uno specifico bot (dato il suo IP) per controllare lo stato del bot, ossia se è inattivo (idle) o se sta effettuando un attacco (attacking) e se si qual'è il suo obbiettivo.
- **attack**: invia delle richieste POST a tutta la botnet per iniziare un attacco DDOS verso l'URL indicato.
- **stop**: invia delle richieste GET a tutta la botnet per fermare l'attacco in corso.
- **email**: invia delle richieste POST a tutta la botnet per far inviare una email da ogni bot a tutti i destinatari presenti nel file JSON passato come parametro. Il contenuto del messaggio viene estrapolato da un file di testo anch'esso passato come parametro.
- **exit**: termina l'esecuzione del centro di comando e controllo.

2.2 Bot (bot.py)

Nel file bot.py viene definita una classe Bot che verrà utilizzata come handler delle richieste HTTP. Una volta creato il server HTTP farà uso di questa classe per rispondere alle varie richieste del C&C rimanendo in ascolto su una porta assegnata dinamicamente dal sistema operativo. In questa maniera sarà più difficile per gli utenti dei sistemi zombie individuare il processo del bot in esecuzione, che inoltre non avrà bisogno di permessi di root per utilizzare le well-known port.

All'avvio il bot si connette alla socket del C&C per comunicargli il suo indirizzo IP e la porta su cui è in ascolto il server HTTP per poi chiudere immediatamente la connessione. Allo stesso modo alla chiusura esso si connette alla stessa socket per comunicare al C&C che non sarà più online.

Il bot risponde alle seguenti richieste del C&C:

Richieste GET

- **/info**: il server risponde inviando le informazioni di sistema su cui è in esecuzione il bot quali: il sistema operativo, il nome del nodo, la release del sistema operativo, la versione del sistema operativo e l'architettura della macchina.

- **/status**: il server risponde inviando lo stato della macchina, ossia se è inattivo (idle) o sta compiendo un attacco (attacking) e, nel caso lo stia facendo, l'URL del suo taget.

- **/stop**: il server interrompe l'attacco in corso sul bot se ce n'è uno, altrimenti il server risponde con un codice di errore HTTP. Inoltre imposta lo stato del bot ad inattivo (idle).

Richieste POST

- **/start**: il bot inizia un attacco. Quando il server riceve una richiesta a questo path gli viene fornito nel body della richiesta l'URL da attaccare. Se il bot sta già effettuando un attacco o l'URL da attaccare non è raggiungibile risponde con un codice d'errore HTTP, altrimenti avvia un thread che si occupa di inviare richieste HTTP una dopo l'altra all'URL indicato. Questo thread termina solamente quando un flag viene impostato a true e ciò avviene quando il bot riceve una richiesta al path '/stop'. Precedentemente all'avvio del thread viene cambiato lo stato del bot da 'Idle' ad 'Attacking' e il suo target viene aggiornato alll'URL che il bot sta attaccando.

- **/email**: il bot invia una email ai destinatari indicati. Quando il server riceve una richiesta a questo path gli vengono forniti nel body della richiesta gli indirizzi a cui inviare il messaggio, l'oggetto e il testo del messaggio. L'email viene inviata ai destinatari in copia carbone nascosta in modo che essi non possano vedere che si tratta di un messaggio inviato ad un grande numero di destinatari.

3 Test della botnet

Per verificare il funzionamento della botnet sono state create diverse macchine virtuali con sistema operativo Debian connesse tra di loro tramite una sottorete virtuale.

Alla VM dalla quale viene eseguito il C&C è stato assegnato un indirizzo IP statico in modo tale che possa essere raggiunta dai bot, i quali invece utilizzano il protocollo DHCP per l'assegnamento di un indirizzo. Questo indirizzo verrà comunicato al C&C quando il bot effettua la connessione.

Per eseguire il codice seguire le istruzioni presenti nel file 'hw/README.md'.