Relazione Homework

Pannozzo Alessio, 1960374

Traccia: 1

Introduzione:

Il progetto prevede la creazione di una botnet composta da agenti di attacco(bot) e da un centro di comando e controllo(C&C).
L'utilizzatore del C&C, attraverso un semplice menù, può dire al programma di inviare richieste ad uno o più bot contemporaneamente sui protocolli specificati dal programma in esecuzione sulle macchine compromesse.
Il bot, non attacca le risorse della macchina sulla quale è in esecuzione, ma fa da tramite alle richieste del C&C.

Scelte implementative:

Per entrambe i programmi è stato scelto il linguaggio *Pyhton(3.11.2)*.
I bot utilizzano i protocolli *HTTP* ed *IRC*.
Per l'IRC è stato installato un server(*Inspircd3*) sulla stessa macchina del C&C.

Tipologie di attacco:

I bot, presenti sulle macchine compromesse, sono in grado di:

- eseguire attacchi HTTP di tipo *GET* ad un url specificato dal C&C;
- raccogliere informazioni sul sistema compromesso(*tipologia del sistema*, *nome del nodo*, *release dell'OS*, *versione* ed *architettura*);
- inviare *mail* ad un set di indirizzi specificati dal C&C.

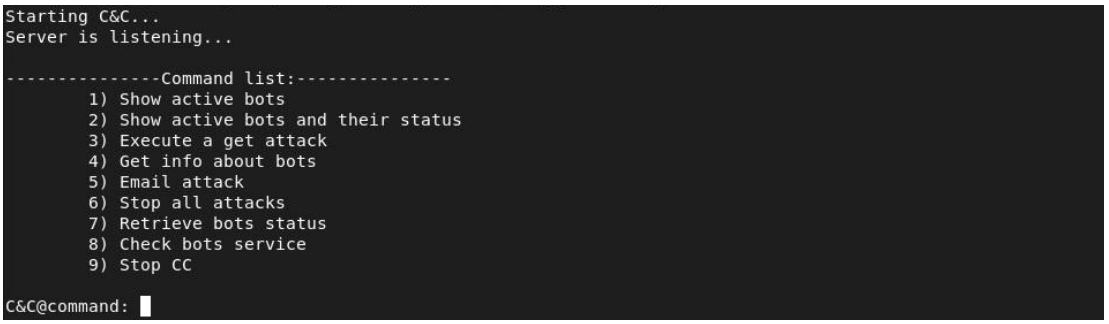
C&C:

Il C&C(*CC.py*) una volta avviato, carica tutti i bot precedentemente attivi memorizzati su un file alla chiusura del programma(activeBot.json) e ne controlla lo stato. Successivamente instanzia un oggetto di tipo Event utile alla terminazione del programma ed avvia due Thread:
Il primo avvia una socket sulla porta 49171 e rimane in ascolto fino alla terminazione del programma gestendo la ricezione delle informazioni da parte dei bot.
Il secondo Thread gestisce le interazioni con l'utilizzatore del programma.
Il C&C consente:

1. la visualizzazione dei servizi disponibili di ogni bot;
2. la visualizzazione dei servizi e dello stato di ogni bot;
3. l'esecuzione di un attacco HTTP;
4. la ricezione di informazioni di sistema dalle macchine compromesse;
5. l'invio di mail ad un set di indirizzi(disponibili nel file email.json);

6. l'arresto di ogni attacco HTTP(*in caso di attacco infinito**);
7. la ricezione di informazioni relative allo stato di ogni bot;
8. il controllo dei servizi offerti da ogni bot;
9. l'arresto del programma attraverso l'oggetto di tipo Event dichiarato in precedenza.

Le voci 3, 4 e 5 consentono all'utente di scegliere il tipo di servizio da utilizzare per effettuare le richieste ad uno o più bot contemporaneamente.
In caso di terminazione del programma, il C&C memorizza lo stato di ogni bot nel file descritto in precedenza.



Avvio del C&C e visualizzazione del menù.

Bot:

Ogni bot(*bot.py*), una volta avviato, manda le proprie informazioni, sulla porta 49171, al C&C quali ip, porta http e porta irc.
Di base, il programma, ha due thread attivi, ma il numero può aumentare in base al numero di attacchi in esecuzione.

```
root        2228  0.2  0.5 250960 21960 pts/0    Sl   15:42   0:00 python bot.py
```

Risorse utilizzate con due thread attivi.

```
root        2228  0.7  0.5 250960 21960 pts/0    Sl   15:42   0:00 python bot.py
```

Risorse utilizzate con più di due thread attivi.

Il primo thread avvia un server *HTTP(httpServerRH)* sulla porta 80:
Per l'implementazione di quest'ultimo è stata usata la libreria http.server di python ed è stato fatto override dei metodi *do_GET* e *do_POST*.
Nel metodo *do_GET* sono stati implementati tre *path*:

- "*/status*": restituisce lo stato del bot;
- "*/stopAttack*": ferma tutti gli attacchi HTTP in corso(anche quelli avviati tramite IRC);
- "*/getSystemInfo*": restituisce le infomazioni del sistema.

Nel metodo *do_POST*, invece, sono stati implementati due *path*

- "*/doGet*": crea un nuovo thread ed esegue un attacco HTTP;
- "*/sendEmail*": invia una mail a tutti gli indirizzi ricevuti dal C&C.

Il path getSystemInfo restituisce le informazioni del bot al C&C, mentre tutti gli altri restituiscono lo stato del sistema.

Il secondo thread avvia una connessione al server IRC presente sulla stessa macchina del C&C, si registra come "*bot-ip(con trattini al posto dei punti)*", e si mette in ascolto. Questo servizio può eseguire le stesse operazioni eseguite dal server HTTP.

Test:

I Test sono stati effettuati su delle vm con sistema operativo Debian 11.
Sono state create: 1 virtual machine per il C&C e 2 virtual machine per i Bot; successivamente, sempre attraverso il programma di virtualizzazione, è stata creata una sottorete NAT che permette alle macchine di comunicare.
Sulla macchina del C&C è stato installato il server IRC e attraverso la configurazione sono stati assegnati dei privileggi al C&C per inviare messaggi in broadcast.

***Attenzione*:** per mostrare il corretto funzionamento del bot, vengono stampati a video tutti i messaggi in arrivo e le operazioni svolte, nel realtà questo non dovrebbe avvenire, per cui tutti i messaggi di log possono essere disabilitati.

Controllo dello stato di ogni bot:

```
C&C@command: 2
Ip              http            irc             target                  action
10.0.2.12       80              6667            -                       waiting
10.0.2.11       80              6667            -                       waiting
```

Bot già attivo e avvio del C&C con relativo controllo dei servizi offerti dai bot memorizzati sul file:

```
(BotNet) root@debian:/home/zznnp/Desktop/BotNet/Bot# 10.0.2.10 - - [22/May/2023
15:21:44] "GET /status HTTP/1.1" 200 -
:cc!cc@127.0.0.1 PRIVMSG bot-10-0-2-12 :#botnet: PING
```

Invio di una richiesta d'attacco HTTP da parte del C&C(protocollo IRC):

```
C&C@command: 3
Select type of attack(1=HTTP, 2=IRC): 2
Enter site url: http://www.youtube.com
Enter number of attack(-1=infinite): -1
Enter target(ip or all): all
```

Ricezione di una richiesta d'attacco dal servizio IRC:

```
:cc!cc@127.0.0.1 PRIVMSG $* :#botnet: get|http://www.youtube.com|-1
Get to http://www.youtube.com
Get to http://www.youtube.com
Get to http://www.youtube.com
Get to http://www.youtube.com
Get to http://www.youtube.com
Get to http://www.youtube.com
Get to http://www.youtube.com
Get to http://www.youtube.com
Get to http://www.youtube.com
Get to http://www.youtube.com
```

Invio di una richiesta d'attacco email da parte del C&C(protocollo HTTP):

```
C&C@command: 5
Select type of attack(1=HTTP, 2=IRC): 1
Enter target(ip or all): all
Sending requests...
Requests sent
```

Ricezione di una richiesta d'attacco email(protocollo IRC):

```
:cc!cc@127.0.0.1 PRIVMSG $* :#botnet: send|Welcome to Our Newsletter - Thank You
 for Signing Up!|I hope this email finds you well. I would like to extend a warm
 welcome and express our heartfelt appreciation for joining our newsletter commu
nity. More info: https://www.youtube.com/watch?v=xvFZjo5PgG0|['gioelezoccoli99@g
mail.com', 'flypilot.51@gmail.com', 'lorensosp0401@gmail.com', 'alessio_pannozzo
@libero.it', 'pannozzo.1960374@studenti.uniroma1.it']
Sending email to 'gioelezoccoli99@gmail.com'
Sending email to  'flypilot.51@gmail.com'
Sending email to  'lorensosp0401@gmail.com'
Sending email to  'alessio_pannozzo@libero.it'
Sending email to  'pannozzo.1960374@studenti.uniroma1.it'
```

Ricezione di uno comando stopAttack sul servizio HTTP:

```
Get to http://www.youtube.com
Get to http://www.youtube.com
Get to http://www.youtube.com
Get to http://www.youtube.com
Get to http://www.youtube.com
Get to http://www.youtube.com
10.0.2.10 - - [24/May/2023 16:14:47] "GET /stopAttack HTTP/1.1" 200 -
Get to http://www.youtube.com
```

Ricezione delle informazioni relative alle macchine compromesse attraverso HTTP:

```
C&C@command: 4
Select type of attack(1=HTTP, 2=IRC): 1
Enter target(ip or all): all
        -----System info about 10.0.2.12 -----
        System:                 Linux
        Node name:              debian
        Release:                5.10.0-23-amd64
        Version:                #1 SMP Debian 5.10.179-1 (2023-05-12)
        Machine:                x86_64
        -----System info about 10.0.2.11 -----
        System:                 Linux
        Node name:              debian
        Release:                5.10.0-23-amd64
        Version:                #1 SMP Debian 5.10.179-1 (2023-05-12)
        Machine:                x86_64
```

Ricezione email: