

## Bot

### Git & Make

U Git repozitorij dodajte poddirektorij lab1 u kojem se treba nalaziti odgovarajuća [Mm]akefile datoteka s definiranim ciljevima:

- *defaultni* cilj je kreiranje izvršnih datoteka bot i UDP\_server,
- *make clean* – obriše izvršne programe i sve ostale nepotrebne datoteke.

Nakon poziva make se kao rezultat moraju dobiti izvršni programi bot i UDP\_server.

### Argumenti i opcije

```
./bot server_ip server_port
```

server\_ip naziv ili IP adresa C&C poslužitelja („Command & Control Server”)

server\_port naziv ili broj UDP *porta* na kojem sluša C&C poslužitelj

```
./UDP_server [-l port] [-p payload]
```

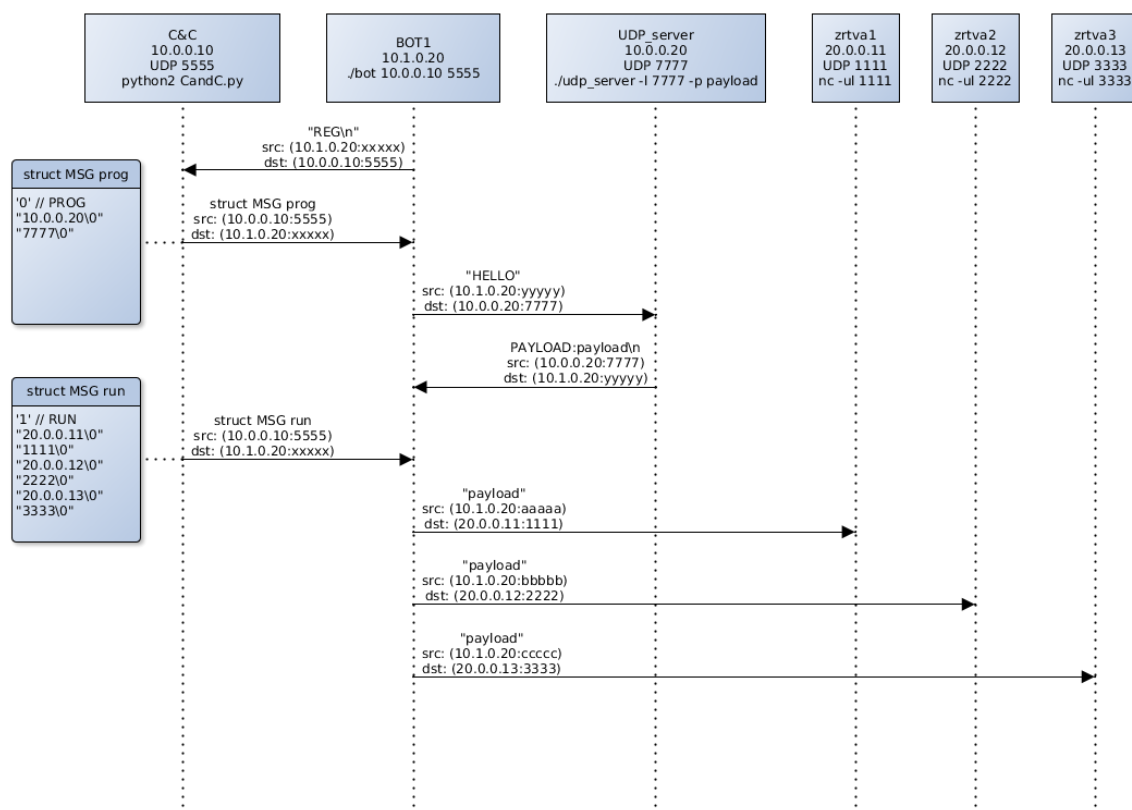
port naziv ili broj UDP *porta* na kojem sluša UDP\_server

payload poruka koju šalje UDP\_server kad primi HELLO od bota

Pretpostavljena vrijednost *porta* port je 1234, dok je poruka payload prazna.

### Botnet

„Botnet” je naziv za skup računala zaraženih zlonamjernim programom koji omogućava udaljenu kontrolu nad zaraženim računalima s tako zvanog „Command & Control” poslužitelja. Korisnik zaraženog računala u pravilu nije svjestan da mu je računalo zaraženo i da sudjeluje u raznim, obično zlonamjernim aktivnostima. Zaraženo računalo postaje tako zvani „zombi” ili „bot” (Web robot) koji prima instrukcije od glavnog računala (engl. Bot master ili C&C server). Poznati *botneti* imali su i nekoliko stotina tisuća *botova*. Nakon početne zaraze, *bot* s C&C poslužitelja dohvaća maliciozni program s kojim će napasti žrtve, te IP adresu jedne ili više žrtava.



### Opis programa

Dohvatite i pokrenite pripremljeni „Command & Control” poslužitelj:

```
$ wget http://mrepro.tel.fer.hr/lab1/CandC.py
```

```
$ python3 CandC.py
```

Program CandC sluša na UDP portu 5555 i očekuje registraciju nekog *bot* klijenta. Također, sa *stdin* prihvaća naredbe *p* i *r* koje registriranom klijentu šalju odgovarajuće naredbe PROG i RUN.

Vaš je zadatak napisati program *bot* koji nakon pokretanja pošalje C&C poslužitelju, na IP adresu *server\_ip* i *port* *server\_port*, UDP datagram sadržaja "REG\n".

Nakon toga sluša poruke od C&C poslužitelja i očekuje poruku u obliku strukture MSG:

```
struct MSG {
    char command
    char IP1[INET_ADDRSTRLEN]
    char PORT1[22]
    ...
    char IP20[INET_ADDRSTRLEN]
    char PORT20[22]
}
```

Struktura MSG sadrži 1 oktet za naredbu, te jedan ili više parova IP adresa i *portova* (maksimalno 20 parova).

Kad *bot* od C&C poslužitelja primi strukturu MSG s poljem *command* jednakim 0 (PROG), u primljenoj strukturi je u prvom idućem zapisu upisan par IP adresa i *port* za UDP\_server koji sluša na navedenim vrijednostima. *Bot* se zatim spaja na dobivenu adresu i UDP *port*, šalje poruku HELLO te učitava poruku duljine najviše 512 znakova. Kad UDP\_server od *bot* klijenta primi poruku HELLO, vraća klijentu poruku veličine do 512 znakova u obliku:

```
PAYLOAD:payload\n
```

Kad *bot* od C&C poslužitelja primi strukturu MSG s poljem *command* jednakim 1 (RUN), u primljenoj strukturi su u idućih N zapisa (maksimalno 20 parova) upisane IP adrese i *portovi* računala koje *bot* napada. Tada *bot* na zadane adrese počinje slati UDP datagrame s porukom *payload* primljenom od UDP\_servera. Poruke šalje periodički, svake sekunde, na sve zadane adrese i to ukupno 15 sekundi. Nakon toga prestaje slati i ponovno čeka poruke od C&C poslužitelja.

Osim *bota*, Vaš je zadatak isprogramirati i UDP poslužitelj UDP\_server tako da sluša na zadanom portu i odgovara na poruku HELLO s nekom porukom.

### Poruke o greškama

Programi moraju provjeravati povratne vrijednosti funkcija i u slučaju problema ispisati poruku na standardni izlaz za greške (*stderr*) te završiti s radom uz izlazni status različit od 0. Ako se programi pozovu s nedefiniranim opcijama treba ispisati poruku:

```
Usage: ./bot server_ip server_port
```

```
Usage: ./UDP_server [-l port] [-p payload]
```

**Primjeri pozivanja:**

U prvom prozoru pokrenite instancu Vašeg poslužiteljskog programa UDP\_server

```
prvi$ ./UDP_server -p payload
```

U drugom prozoru pokrenite pripremljeni „Command & Control” poslužitelj:

```
drugi$ python3 CandC.py
```

U slučaju da se UDP\_server i CandC.py pokreću na odvojenim računalima (npr. pri pokretanju u sustavu IMUNES), kao i u slučaju da se UDP\_server pokreće s *portom* različitim od 1234, tada je prije pozivanja prethodne naredbe u datoteci CandC.py potrebno promijeniti varijable *UDP\_ip* i *UDP\_port* u odgovarajuće vrijednosti.

U primjerima se pretpostavlja da je adresa Vašeg računala 10.0.2.15. (provjerite naredbom `ip a`)

U trećem, četvrtom i petom prozoru pokrenite tri instance Vašeg *bot* klijent programa bot i spojite se na C&C poslužitelj:

```
treći$ ./bot 127.0.0.1 5555
četvrti$ ./bot 10.0.2.15 5555
peti$ ./bot 10.0.2.15 5555
```

Klijenti ostaju registrirani do zaustavljanja „Command & Control” poslužitelja. Prije novog testiranja obavezno zaustavite i ponovo pokrenite program CandC.py.

Na „Command & Control” poslužitelju provjerite aktivne *botove*:

```
C&C> l
-> lista botova:
10.0.2.15:10525; 127.0.0.1:20781; 10.0.2.15:37694;
```

Pokrenite Wireshark i snimajte razmjenu poruka između C&C poslužitelja i *botova*.

Svim *botovima* pošaljite naredbu PROG:

```
C&C> p
```

U novom prozoru pokrenite NetCat koji će glumiti žrtvu kojoj *botovi* šalju UDP poruke:

```
peti$ nc -ku -l -p 5678
```

Svim *botovima* pošaljite naredbu RUN:

```
C&C> r
```

NetCat ispisuje primljene poruke.

Ispis svih podržanih naredbi:

```
C&C> h
Podržane su naredbe:
p ... Bot klijentima šalje poruku PROG
    struct MSG:0 127.0.0.1 1234
r ... Bot klijentima šalje poruku RUN s adresama iz ifconfig
    struct MSG:1 10.0.2.15 5678 192.168.56.101 6789 127.0.0.1 hostmon
r2... Bot klijentima šalje poruku RUN s nekim adresama
    struct MSG:1 20.0.0.11 1111 20.0.0.12 2222 20.0.0.13 3333
l ... lokalni ispis adresa bot klijenata
n ... šalje poruku: NEPOZNATA
q ... zavrsetak rada programa
h ... ispis naredbi
```