

Bot

Git & Make

U Git repozitorij dodajte poddirektorij lab2 u kojem se treba nalaziti odgovarajuća [Mm]akefile datoteka s definiranim ciljevima:

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

Nakon poziva make se kao rezultat moraju dobiti izvršni programi bot i server.

Argumenti i opcije

`./bot ip port`

`ip` naziv ili IP adresa C&C poslužitelja („Command & Control Server”)
`port` naziv ili broj UDP *porta* na kojem sluša C&C poslužitelj

`./server [-t tcp_port] [-u udp_port] [-p popis]`

`tcp_port` naziv ili broj TCP *porta* na kojem sluša C&C poslužitelj
`udp_port` naziv ili broj UDP *porta* na kojem sluša C&C poslužitelj
`popis` popis *payloadova* odvojenih dvotočkom (:) koje će program bot slati žrtvama

Pretpostavljena vrijednost *portova* je 1234, dok je popis *payloadova* prazan.

U drugoj laboratorijskoj vježbi će se postojeći programi iz prve laboratorijske vježbe, bot i UDP_server, modificirati i proširiti novim funkcionalnostima.

„Command & Control” poslužitelj

Dohvatite i pokrenite pripremljeni (novi) „Command & Control” poslužitelj:

```
$ wget http://mrepro.tel.fer.hr/lab2/CandC.py
```

```
$ python3 CandC.py
```

Program CandC sluša na UDP portu 5555 i očekuje registraciju jednog ili više *bot* klijenata. Također, sa *stdin* prihvaća naredbe pt, pu, r, s i q koje registriranom klijentu šalju odgovarajuće naredbe PROG_TCP, PROG_UDP, RUN, STOP i QUIT.

Opis programa bot

Vaš je zadatak proširiti postojeći program bot iz 1. laboratorijske vježbe na način opisan u nastavku. Program bot nakon pokretanja pošalje C&C poslužitelju, na ip:port, UDP datagram sadržaja "REG\n".

Nakon toga konstantno 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 od 0 do najviše 20 parova IP adresa i *portova* koji mogu biti zapisani kao naziv ili brojčano. Među IP adresama se može nalaziti i *broadcast* adresa.

Kad *bot* od C&C poslužitelja primi strukturu MSG s poljem command jednakim 0 (QUIT) program bot prestaje s radom.

Kad *bot* od C&C poslužitelja primi strukturu MSG s poljem command jednakim 1 (PROG_TCP), spaja se na TCP poslužitelj na IP adresi i *portu* zapisanim u prvom sljedećem zapisu, šalje poruku "HELLO\n", učitava odgovor duljine najviše 1024 znaka te zatvara TCP konekciju.

Kad *bot* od C&C poslužitelja primi strukturu MSG s poljem `command` jednakim 2 (PROG_UDP), na IP adresu i UDP *port* primljene u prvom idućem zapisu; šalje poruku "HELLO\n" te učitava odgovor duljine najviše 1024 znaka.

Kad program server od *bot* klijenta primi TCP ili UDP poruku "HELLO\n", na odgovarajući TCP ili UDP *port* mu vraća poruku veličine do 1024 znaka oblika:

```
payload1:payload2:...:payloadN:\n
```

Kad *bot* od C&C poslužitelja primi strukturu MSG s poljem `command` jednakim 3 (RUN), u primljenoj strukturi su u idućih *M* zapisa (maksimalno 20 parova) upisane IP adrese i *portovi* računala koje *bot* napada. Tada *bot* na zadane adrese počinje slati UDP datagrame s porukama *payload1*, *payload2*, ..., *payloadN* primljenima od programa server. Prolazi po dobivenom popisu *N payloadova* te svaki od njih šalje jednom na svaku od zadanih *M* adresa. Svih *M*N* poruka ponovno šalje svake sekunde, maksimalno 100 sekundi, ili ako se na neki od dalje navedenih načina zaustavi slanje.

Kad bilo koja „žrtva” vrati neki podatak *botu* on prestaje sa slanjem poruka svim „žrtvama”.

Kad *bot* od C&C poslužitelja primi strukturu MSG s poljem `command` jednakim 4 (STOP) program *bot* prestaje sa slanjem poruka žrtvama.

Opis programa server

Osim *bota*, Vaš je zadatak isprogramirati i TCP/UDP poslužitelj server tako da sluša na zadanim *portovima* i odgovara na poruke "HELLO\n" s nekom porukom. Poruka je oblika:

```
payload1:payload2:...:payloadN:\n
```

Program server na kraj popisa dodaje oznaku za kraj reda (\n). Takvu poruku šalje na TCP odnosno UDP *port* s kojeg je dobio poruku "HELLO\n".

Također omogućite programu server slušanje na standardnom ulazu (*stdin*) gdje očekuje poruke PRINT, SET i QUIT.

Kad program server na *stdin* primi poruku PRINT, na standardni izlaz (*stdout*) ispisuje trenutno spremljeni popis *payloadova*.

Kad program server na *stdin* primi poruku SET novi_popis, popis *payloadova* postavlja na "novi_popis".

Kad program server na *stdin* primi poruku QUIT, prestaje s radom uz izlazni status jednak 0.

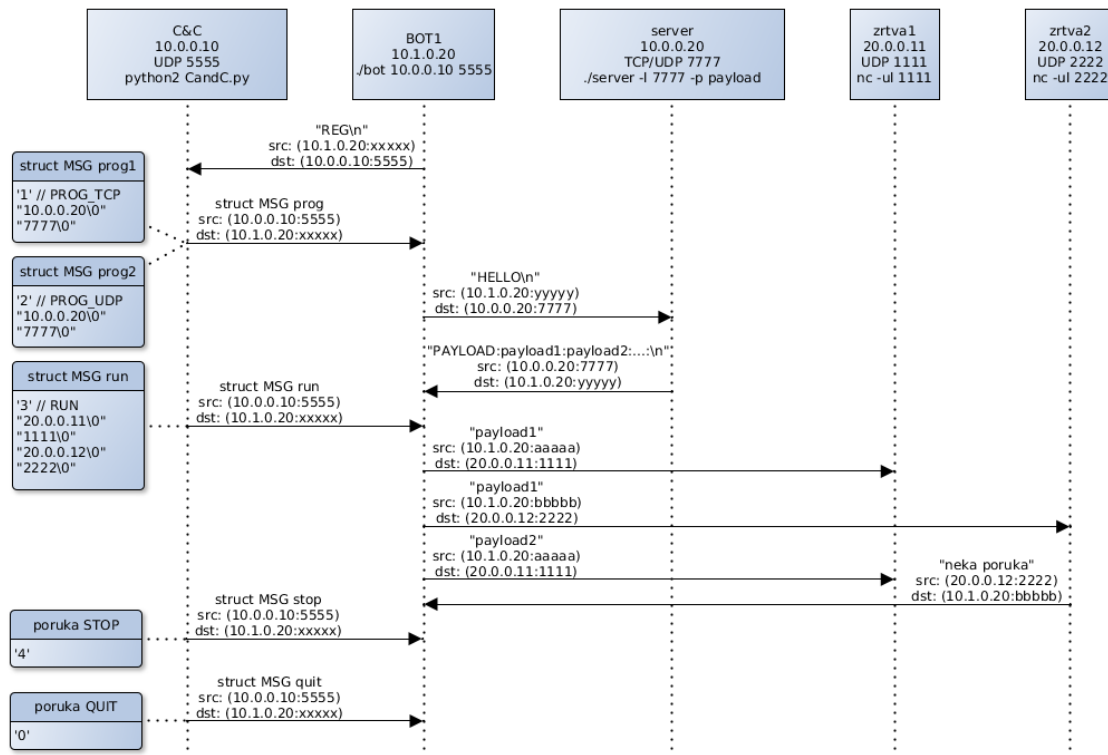
Multipleksiranje ulazno/izlaznih operacija u programu server izvedite uz pomoć funkcija `select` ili `poll` bez korištenja procesa ili dretvi.

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 ip port
```

```
Usage: ./server [-t tcp_port] [-u udp_port] [-p popis]
```



Primjeri pozivanja:

U prvom prozoru pokrenite instancu vašeg poslužiteljskog programa server

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

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

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

U slučaju da se server i CandC.py pokreću na odvojenim računalima (npr. pri pokretanju u sustavu IMUNES), kao i u slučaju da se 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 `ifconfig`)

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
```

```
cetvrti$ ./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> pu
```

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

```
peti$ nc -ul -p 5678
```

Svim *botovima* pošaljite naredbu RUN:

```
C&C> r
```

NetCat ispisuje primljene poruke. Kad „žrtva” (NetCat klijent) vrati neki podatak *botu* on prestaje sa slanjem poruka.

Ispis svih podržanih naredbi:

```
C&C> h
```

Podržane su naredbe:

```
pt... Bot klijentima ssalje poruku PROG_TCP
```

```
struct MSG:1 127.0.0.1 1234
```

```
pu... Bot klijentima ssalje poruku PROG_UDP
```

```
struct MSG:2 127.0.0.1 1234
```

```
r ... Bot klijentima ssalje poruku RUN s adresama iz ifconfig
```

```
struct MSG:3 10.0.2.15 vat 192.168.56.101 6789 127.0.0.1 prosharerequest
```

```
r2... Bot klijentima ssalje poruku RUN s nekim adresama
```

```
struct MSG:3 20.0.0.11 1111 20.0.0.12 2222 20.0.0.13 dec-notes
```

```
s ... Bot klijentima ssalje poruku STOP ('4')
```

```
l ... lokalni ispis adresa bot klijenata
```

```
n ... salje poruku: NEPOZNATA
```

```
q ... Bot klijentima ssalje poruku QUIT ('0') i završava s radom
```

```
h ... ispis naredbi
```