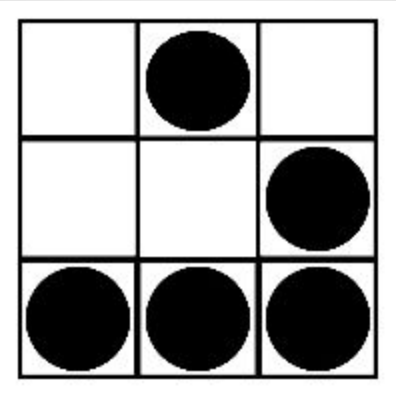
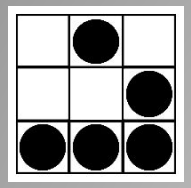
Perl в хэке и хэки в Perl

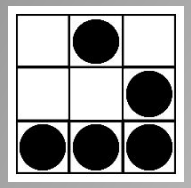


Докладчик:
Илья Зеленчук,
Perlclub УрГУ (г. Екатеринбург)



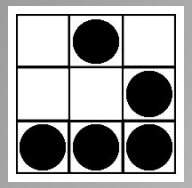
Игры

- CTF (Capture the Flag)
- ICFP
- EACP (Extremely Advanced Computer Programming)

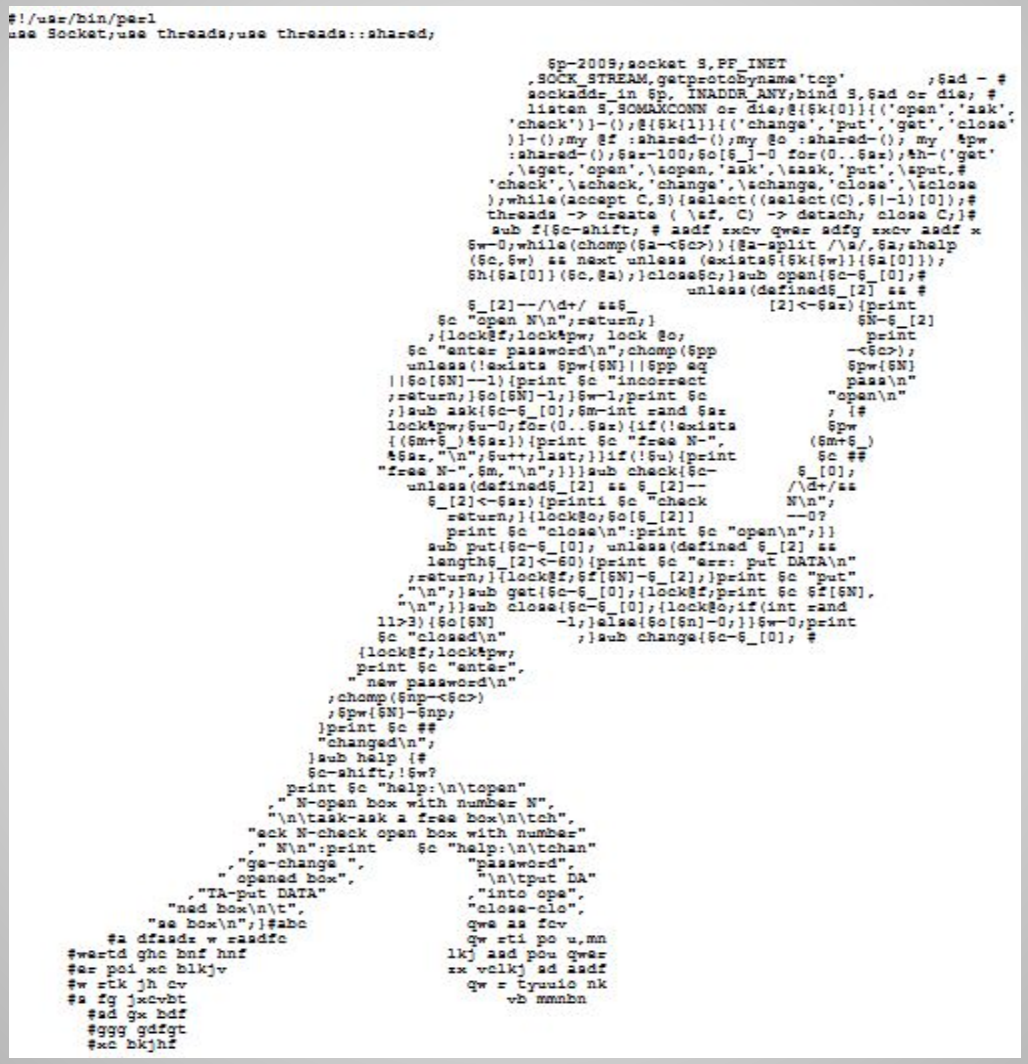


Perl в хэке

- Простоя работа с сетью;
- Удобен при написании PoC;
- Обфускация кода;
- Генерация сложночитаемого C кода;
- Затрудненный reverse.

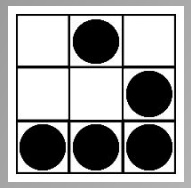


Пример игрового сервиса



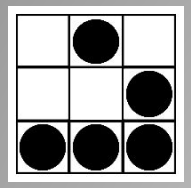
```
#!/usr/bin/perl
use Socket;use threads;use threads::shared;

                                          $p=2009;socket S,PF_INET
                                        ,SOCK_STREAM,getprotobyname'tcp'        ;$ad = #
                                        sockaddr_in $p, INADDR_ANY;bind S,$ad or die; #
                                        listen S,SOMAXCONN or die;@{$k{0}}{('open','ask',
                                      'check')}=();@{$k{1}}{('change','put','get','close'
                                      )}=();my @f :shared=();my @o :shared=(); my  %pw
                                      :shared=();$sx=100;$o[$_]=0 for(0..$sx);%h=('get'
                                      ,\&get,'open',\&open,'ask',\&ask,'put',\&put,#
                                    'check',\&check,'change',\&change,'close',\&close
                                    );while(accept C,S){select((select(C),$|=1)[0]);#
                                    threads -> create ( \&f, C) -> detach; close C;}#
                                     sub f{$c=shift; # asdf zxcv qwer sdfg zxcv asdf x
                                  $w=0;while(chomp($a=<$c>)){@a=split /\s/,$a;&help
                                  ($c,$w) && next unless (exists${$k{$w}}{$a[0]});
                                  $h{$a[0]}($c,@a);}close$c;}sub open{$c=$_[0];#
                                                        unless(defined$_[2] && #
                                    $_[2]--/\d+/ &&$_                [2]<-$sx){print
                                  $c "open N\n";return;}                      $N-$_[2]
                                 ;{lock@f;lock%pw; lock @o;                    print
                                $c "enter password\n";chomp($pp                -<$c>);
                                unless(!exists $pw{$N}||$pp eq                 $pw{$N}
                              ||$o[$N]--1){print $c "incorrect                 pass\n"
                              ;return;}$o[$N]-1;}$w=1;print $c                "open\n"
                              ;}sub ask{$c=$_[0];$m=int rand $sx               ; {#
                              lock%pw;$u=0;for(0..$sx){if(!exists              $pw
                              {($m+$_)%$sx]}{print $c "free N-",              ($m+$_)
                              %$sx,"\n";$u++;last;}}if(!$u){print               $c ##
                            "free N-",$m,"\n";}}}sub check{$c=                 $_[0];
                              unless(defined$_[2] && $_[2]--                  /\d+/&&
                                $_[2]<-$sx){printi $c "check                  N\n",
                                  return;}{lock@o;$o[$_[2]]                   --0?
                                  print $c "close\n":print $c "open\n";}}
                                sub put{$c=$_[0]; unless(defined $_[2] &&
                                length$_[2]<-60){print $c "err: put DATA\n"
                               ;return;}{lock@f;$f[$N]-$_[2];}print $c "put"
                               ,"\n";}sub get{$c=$_[0];{lock@f;print $c $f[$N],
                                "\n";}}sub close{$c=$_[0];{lock@o;if(int rand
                             ll>3){$o[$N]      -1;}else{$o[$n]=0;}}$w=0;print
                              $c "closed\n"       ;}sub change{$c=$_[0]; #
                            {lock@f;lock%pw;
                            print $c "enter",
                           " new password\n"
                          ;chomp($np=<$c>)
                          ;$pw{$N}=$np;
                         }print $c ##
                         "changed\n";
                        }sub help {#
                        $c=shift;!$w?
                      print $c "help:\n\topen"
                     ," N-open box with number N",
                    "\n\task-ask a free box\n\tch",
                  "eck N-check open box with number"
                 ," N\n":print    $c "help:\n\tchan"
               ,"ge-change ",          "password",
             " opened box",             "\n\tput DA"
           ,"TA-put DATA"              ,"into ope",
         "ned box\n\t",                "close-clo",
       "se box\n";}#abc                qwe as fcv
    #a dfasdx w zasdfc                 qw zti po u,mn
#wertd ghc hnf hnf                    lkj asd pou qwer
#er poi xc blkjv                      zx vclkj ad zsdf
#w rtk jh cv                            qw = tyuuio nk
#a fg jxc-vbt                               vb mnnbn
  #ad gx bdf
  #ggg gdfgt
  #xc hkjhf
```

YAPC::Russia 2009



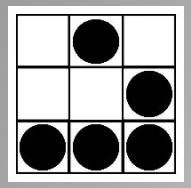
Простой веб клиент

;

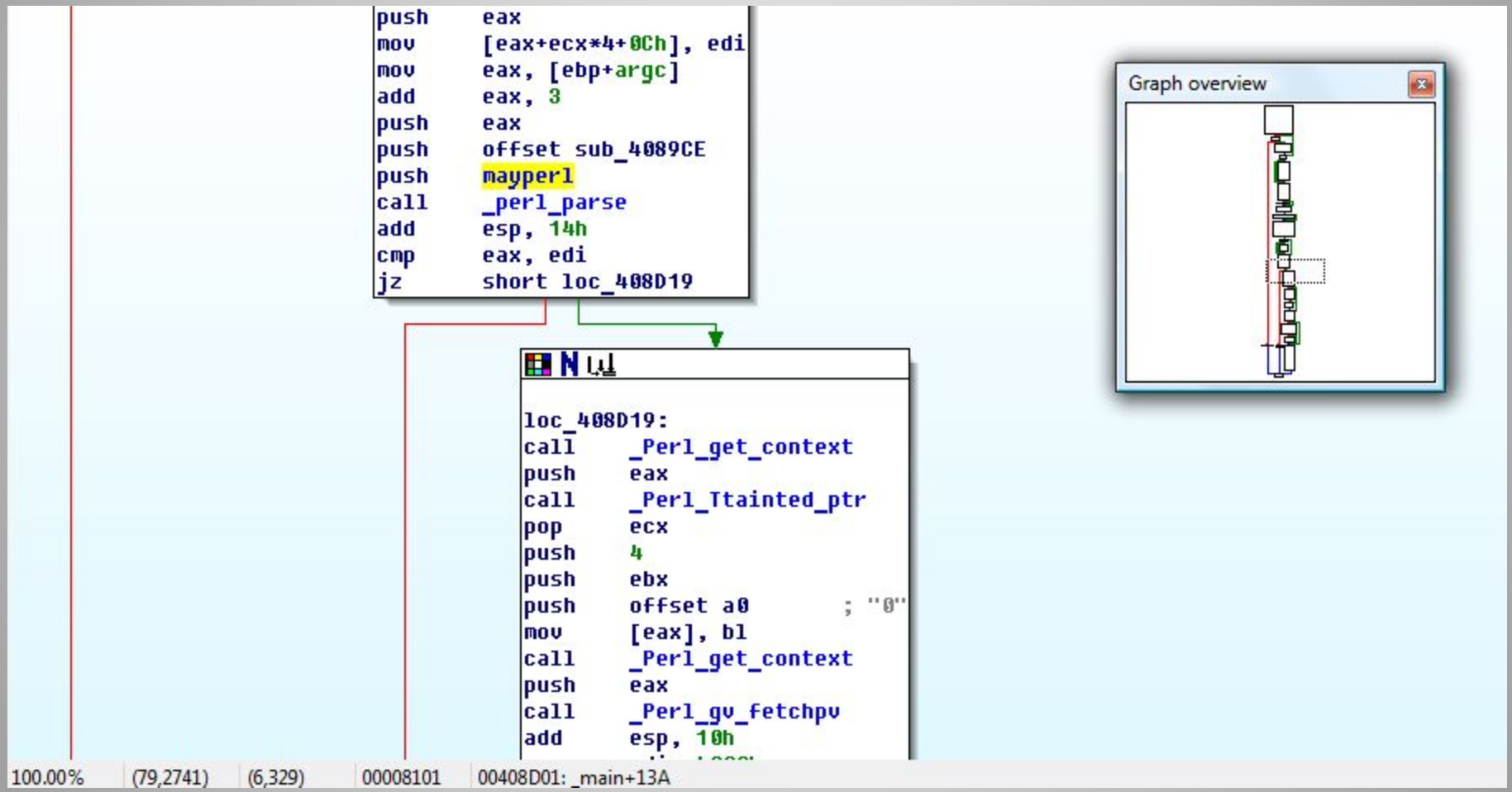


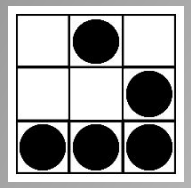
Perl2C

```
... (3474 строки)
xpv_list[79].xpv_pv = savepvn("Hello, MayPerl\n", 15);
{
        SV **svp;
        AV *av = (AV*)&sv_list[279];
        av_extend(av, 2);
        svp = AvARRAY(av);
        *svp++ = (SV*)&PL_sv_undef;
        *svp++ = (SV*)&PL_sv_undef;
        *svp++ = (SV*)&sv_list[280];

        AvFILLp(av) = 2;
}
PL_curpad = AvARRAY((AV*)&sv_list[279]);
GvHV(PL_incgv) = (HV*)&sv_list[53];
... (150 строк)
```



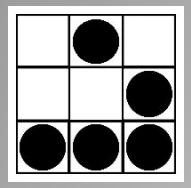
Perl2bin


push eax
mov [eax+ecx*4+0Ch], edi
mov eax, [ebp+argc]
add eax, 3
push eax
push offset sub_4089CE
push mayperl
call _perl_parse
add esp, 14h
cmp eax, edi
jz short loc_408D19
loc_408D19:
call _Perl_get_context
push eax
call _Perl_Ttainted_ptr
pop ecx
push 4
push ebx
push offset a0 ; "0"
mov [eax], bl
call _Perl_get_context
push eax
call _Perl_gv_fetchpv
add esp, 10h
Graph overview
100.00% (79,2741) (6,329) 00008101 00408D01: _main+13A



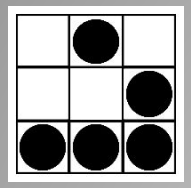
Хэки в Perl

- Простой сокет в Perl’e;
- Sniffer под UNIX без использования libpcap;
- Прием/отправка пакетов и использованием raw socket;
- Неблокирующие сокеты.



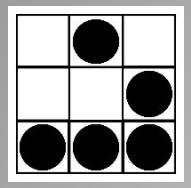
Perl sockets

```
#!/usr/bin/perl

use Socket;

socket $S, PF_INET, SOCK_STREAM, getprotobyname('tcp‘);
my $addr=sockaddr_in(80, inet_aton($ip));
connect $S, $addr or die "Can't open connection: $!\n";

send $S, "GET / HTTP/1.0\r\n\r\n", 0;
print  <$S>;

close $S;
```



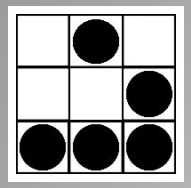
Sniffer под Unix без использования libpcap

```
#!/usr/bin/perl

#use Socket;
use constant PF_PACKET => 17;
use constant SOCK_PACKET => 10;
use constant ETH_P_ALL => 0x0003;

socket (SOCKET, PF_PACKET, SOCK_PACKET, ETH_P_ALL) or die “Socket error: $!\n";
while (){
        recv (SOCKET, $buf, 1514, 0);          # читаем пакет
        print unpack ("H*", $buf), "\n\n";  # выводим его в формате hex
}
```



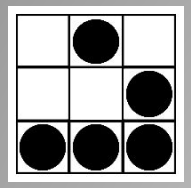
Отправка UDP пакета через raw socket

```
#!/usr/local/bin/perl

use Socket;
use constant IPPROTO_RAW => 255;

$iaddr = inet_aton ('192.168.139.1');
$paddr = sockaddr_in (80, $iaddr);        #80 - порт назначения

socket(SOCKET, PF_INET, SOCK_RAW, IPPROTO_RAW) or die “Socket error: $!\n";

…
```



Отправка UDP пакета через raw socket

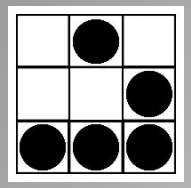
```
$packet .= pack("C", 69);
$packet .= pack ("H2", '00');
$packet .= pack ("n", 28);
$packet .= pack ("n", 0);
$packet .= pack ("H4", '4000');
$packet .= pack ("C", 64);
$packet .= pack ("C", getprotobyname('udp'));
$packet .= pack ("n", 0);
$source_ip = '207.46.197.32';
$result_source_ip .= pack ("C", $_)  for (split('\.', $source_ip));
$packet .= $result_source_ip;
$destination_ip = '192.168.139.1';
$result_destination_ip .= pack ("C", $_) for (split('\.', $destination_ip));
$packet .= $result_destination_ip;
$packet .= pack ("n", 25);
$packet .= pack ("n", 80);
$packet .= pack ("n", 8);
$packet .= pack ("H4", '0000');
```



Отправка пакетов через packet socket

С какого интерфейса происходит отправка пакета:

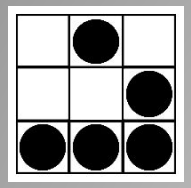
```
struct sockaddr {
        sa_family_t    sa_family;      //семейство протоколов
        char           sa_data[14];    //14 байтов на описание этого семейства...
};
```



Отправка пакетов через packet socket

Пример заполнения структуры и отправки пакета:

```
$addr = PF_PACKET; #семейство
$iface = "eth0"; #используемое устройство
$socket = pack ('Sa14', $addr, $iface); #упаковываем все это в структуру
send(SOCKET, $packet, 0, $socket) or die "Can't send packet:$!\n";
```



Неблокирующий сокет на Perl (Windows)

…

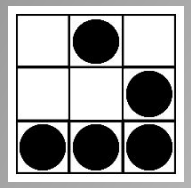
```
my ($win, $ein);
my $addr=sockaddr_in(86, inet_aton("10.0.0.253"));
socket SOCK, PF_INET, SOCK_STREAM, 0 or die "Socket: $!\n";

ioctl(SOCK, 0x8004667e, pack("I", 1));
connect SOCK, $addr;

vec ($win = '',fileno(SOCK),1)=1;
$ein=$win;
my $nfound = select (undef, $win, $ein, 1);
```

…

Perl в хэке и хэки в Perl

СПАСИБО ЗА ВНИМАНИЕ!

Илья Зеленчук
(ilya@hackerdom.ru)



K.I.S.S.

Запустить netcat, повесить bash, cat’нуть файл, grep’нуть по регвыру...

Или лучше установить Perl?