Stegano 100

CamelCase

Задание

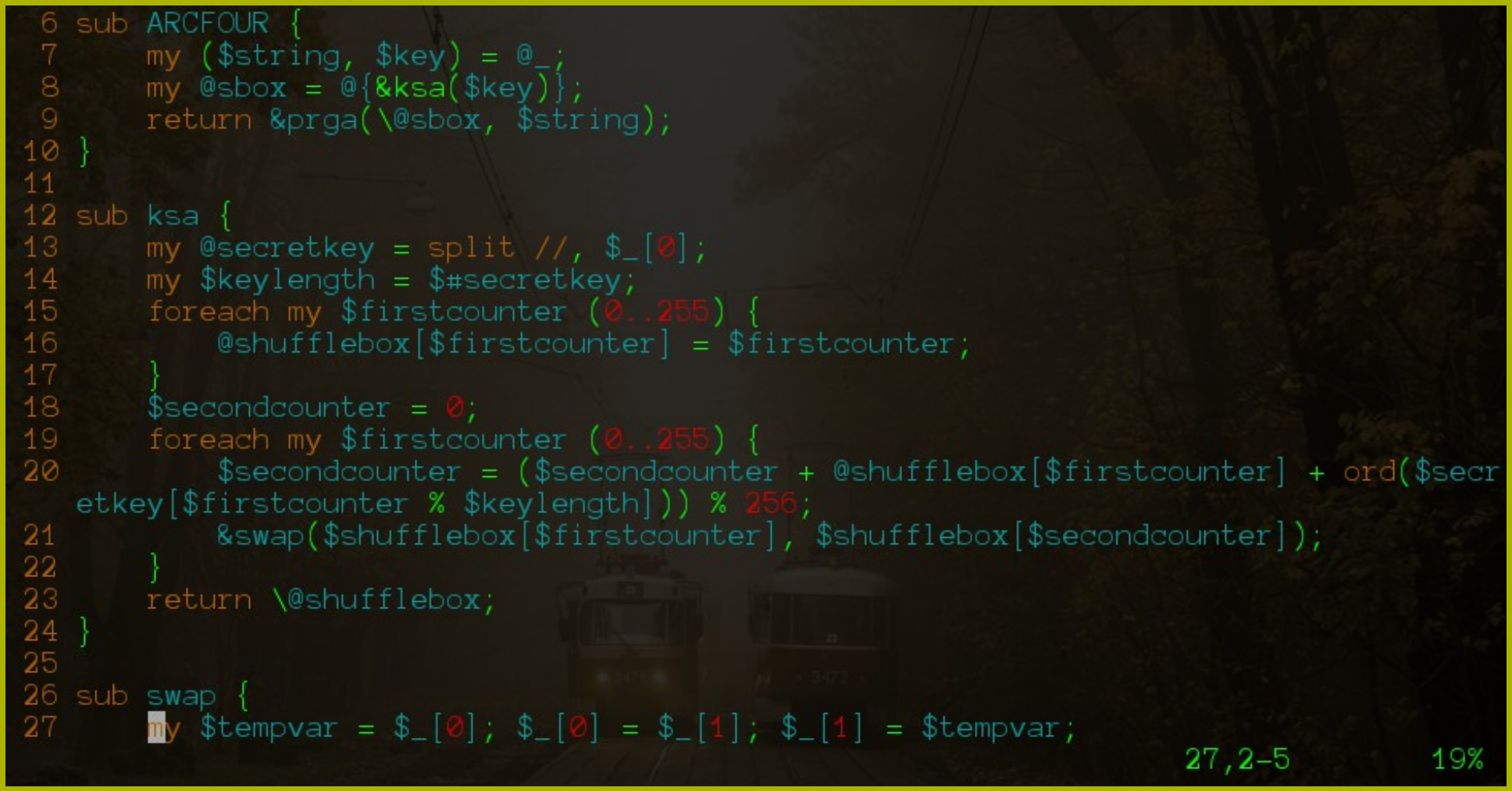
В ходе опративных мероприятий возле консульства был задержан странный субъект, в его личных вещах было обнаружено запоми нающее устройство с программой. Но эта программа ни чего не делает, что крайне странно. Вы, как знающий человек, не подс кажете нам секрет этой программы?

```
1 sub ksa{@SECrETKeY=split//,$_[0];$kEyLENgtH=6;foreach $firStcountER(0..255){@shUF
flEBoX[$firStcountER]=$firStcountER;}$SecoNDcounTeR=0;foreach $firStcountER(0..25
5){$SecoNDcounTeR=($SecoNDcounTeR+@shUFflEBoX[$firStcountER]+ord($SECrETKeY[$firS
tcountER % $kEyLENgtH]))%256;&swap($shUFflEBoX[$firStcountER],$shUFflEBoX[$SecoND
counTeR]);}return \@shUFflEBoX;}$NotOBVioUskey = 'kotenok';sub swap{$teMPvAr=$_[0
];$_[0]=$_[1];$_[1]=$teMPvAr;}$sTr = '55E4ED9435CE1AA0476CF7EF7839F0837751862F074
E78F1A15D841374';sub ARCFOUR{($sTrINg,$kEy)=@_;@SBoX=@{&ksa($kEy)};return&prga(\@
SBoX,$sTrINg);}sub prga{@SSHufFLebOX=@{$_[0]};@TeXt=$_[1]=~/.{1,2}/g;foreach(0..2
8){$TeXt[$_]=hex($TeXt[$_]);}$TeXt_len=28;$ffIrstcounTEr=0;$sSecoNDcoUNtEr=0;$stR
CouNter=0;while($stRCouNter<=$TeXt_len){$ffIrstcounTEr=($ffIrstcounTEr+1)%256;$sS
ecoNDcoUNtEr=($sSecoNDcoUNtEr+$SSHufFLebOX[$ffIrstcounTEr])%256;&swap($SSHufFLebO
X[$ffIrstcounTEr],$SSHufFLebOX[$sSecoNDcoUNtEr]);$TEmpcOUnteR=($SSHufFLebOX[$ffIr
stcounTEr]+$SSHufFLebOX[$sSecoNDcoUNtEr])%256;$cuRReNtkey=@SSHufFLebOX[$TEmpcOUnt
eR];$tempstring.=chr(($cuRReNtkey^$TeXt[$stRCouNter])%256);$stRCouNter++;}return
$tempstring;}
```

Участники получают исполняемый Perl скрипт, написанный в своеобразном стиле:

- он слегка обфусцирован (просто, удалены переносы строк, поэтому весь скрипт единая строка);
- чередование регистров символов весьма странное, хотя в общей каше Perlстиля это может не показаться самой странной вещью.

Деобфусцируем:



```
 6 sub ARCFOUR {
 7     my ($string, $key) = @_;
 8     my @sbox = @{&ksa($key)};
 9     return &prga(\@sbox, $string);
10 }
11
12 sub ksa {
13     my @secretkey = split //, $_[0];
14     my $keylength = $#secretkey;
15     foreach my $firstcounter (0..255) {
16         @shufflebox[$firstcounter] = $firstcounter;
17     }
18     $secondcounter = 0;
19     foreach my $firstcounter (0..255) {
20         $secondcounter = ($secondcounter + @shufflebox[$firstcounter] + ord($secr
   etkey[$firstcounter % $keylength])) % 256;
21         &swap($shufflebox[$firstcounter], $shufflebox[$secondcounter]);
22     }
23     return \@shufflebox;
24 }
25
26 sub swap {
27     my $tempvar = $_[0]; $_[0] = $_[1]; $_[1] = $tempvar;
                                                  27,2-5        19%
```

По совокупности внешних признаков «опознаём» RC4 (или не «опознаём») и добавляем вызов функции расшифровки:

```
 1 my $str = '55E4ED9435CE1AA0476CF7EF7839F0837751862F074E78F1A15D841374';
 2 my $notobviouskey = 'kotenok';
 3
 4 print "->".&ARCFOUR($str, $notobviouskey)."<-\n";
 5
 6 sub ARCFOUR {
 7     my ($string, $key) = @_;
 8     my @sbox = @{&ksa($key)};
 9     return &prga(\@sbox, $string);
10 }
11
12 sub ksa {
13     my @secretkey = split //, $_[0];
14     my $keylength = $#secretkey;
15     foreach my $firstcounter (0..255) {
16         @shufflebox[$firstcounter] = $firstcounter;
17     }
18     $secondcounter = 0;
19     foreach my $firstcounter (0..255) {
20         $secondcounter = ($secondcounter + @shufflebox[$firstcounter] + ord($secr
   etkey[$firstcounter % $keylength])) % 256;
21         &swap($shufflebox[$firstcounter], $shufflebox[$secondcounter]);
22     }
                                                          1,5           Top
```

После удачной дешифровки имеем подсказку:

И, присматриваясь к переменным, понимем, что стеганосообщение скрыто в регистрах символов, составляющих переменных!

Дешифровка

Сопоставляем сиволам, из которых состоят переменные, “0” и “1” в зависомости от регистра.

A => 1

a => 0

Написав достаточно тривиальный скрипт, получаем результат: