Ghidra SRE Final Challenge

Clayton B. Hodges

Colorado Mesa University

CSCI 420: Software Security

Contents

1. Task Completion

1.1 Instructions

I. Install Ghidra, VirtualBox, and materials.

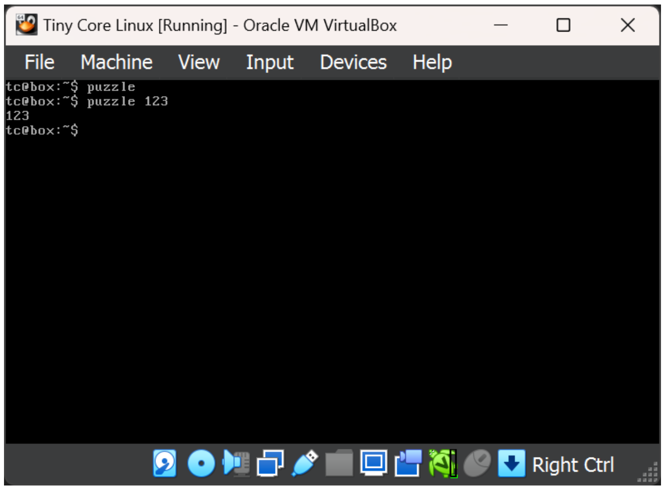
II. Launch Tiny Core Linux and test executable(s) inside.

III. Launch Ghidra, create a project, and upload executable(s).

IV. Use Ghidra to deconstruct the executable(s), break down binary functions, and discover secrets.

1.2 Puzzle

I. Entering ‘puzzle’ and ‘puzzle 123’ in Tiny Core Linux.

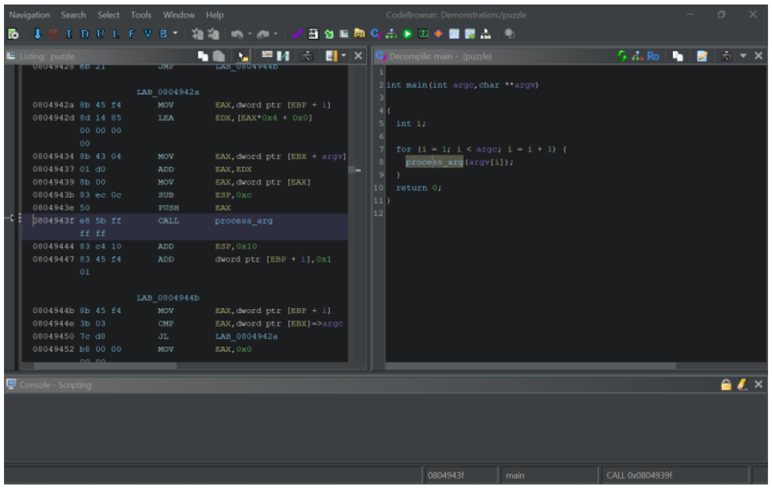


```
tc@box:~$ puzzle
tc@box:~$ puzzle 123
123
tc@box:~$
```

(Puzzle’s expected output when run directly is nothing, but it prints any args it’s passed.)

GHIDRA SRE FINAL CHALLENGE

II. Investigating Puzzle’s deconstructed contents and relabeling content in Ghidra.

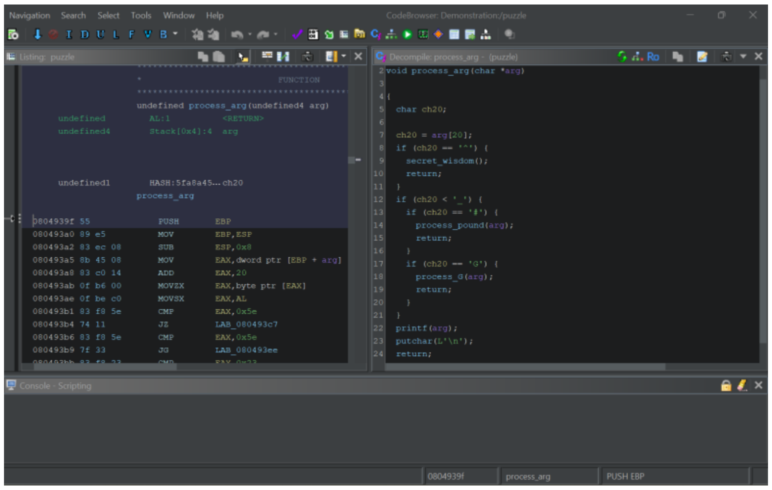

Navigation Search Select Tools Window Help
CodeBrowser: Demonstration:/puzzle
Listing: puzzle
Decompile: main - (puzzle)
Console - Scripting
0804943f
main
CALL 0x0804939f

```
                              LAB_0804942a
0804942a 8b 45 f4        MOV        EAX,dword ptr [EBP + i]
0804942d 8d 14 85        LEA        EDX,[EAX*0x4 + 0x0]
         00 00 00
         00
08049434 8b 43 04        MOV        EAX,dword ptr [EBX + argv]
08049437 01 d0           ADD        EAX,EDX
08049439 8b 00           MOV        EAX,dword ptr [EAX]
0804943b 83 ec 0c        SUB        ESP,0xc
0804943e 50              PUSH       EAX
0804943f e8 5b ff        CALL       process_arg
         ff ff
08049444 83 c4 10        ADD        ESP,0x10
08049447 83 45 f4        ADD        dword ptr [EBP + i],0x1
         01

                              LAB_0804944b
0804944b 8b 45 f4        MOV        EAX,dword ptr [EBP + i]
0804944e 3b 03           CMP        EAX,dword ptr [EBX]=>argc
08049450 7c d8           JL         LAB_0804942a
08049452 b8 00 00        MOV        EAX,0x0
```

```c
int main(int argc,char **argv)

{
  int i;

  for (i = 1; i < argc; i = i + 1) {
    process_arg(argv[i]);
  }
  return 0;
}
```

(The main entry point of the program calls a function we labeled process_arg.)

III. Further investigating Puzzle functions.

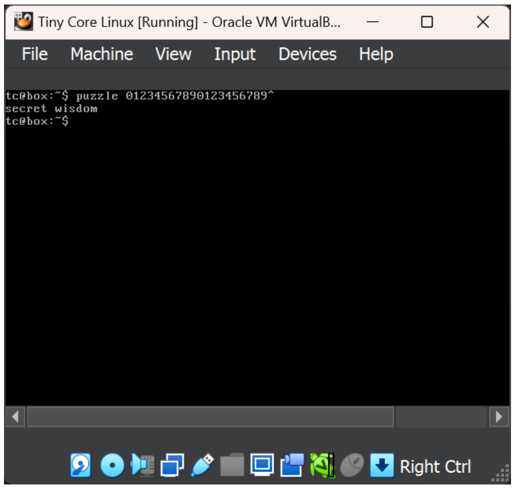

Navigation Search Select Tools Window Help
CodeBrowser: Demonstration:/puzzle
Listing: puzzle
Decompile: process_arg - (puzzle)
Console - Scripting
0804939f
process_arg
PUSH EBP

```
                              *                          FUNCTION
                              ****************************************
                              undefined process_arg(undefined4 arg)
     undefined         AL:1           <RETURN>
     undefined4        Stack[0x4]:4   arg


     undefined1        HASH:5fa8a45... ch20
                              process_arg

0804939f 55              PUSH       EBP
080493a0 89 e5           MOV        EBP,ESP
080493a2 83 ec 08        SUB        ESP,0x8
080493a5 8b 45 08        MOV        EAX,dword ptr [EBP + arg]
080493a8 83 c0 14        ADD        EAX,20
080493ab 0f b6 00        MOVZX      EAX,byte ptr [EAX]
080493ae 0f be c0        MOVSX      EAX,AL
080493b1 83 f8 5e        CMP        EAX,0x5e
080493b4 74 11           JZ         LAB_080493c7
080493b6 83 f8 5e        CMP        EAX,0x5e
080493b9 7f 33           JG         LAB_080493ee
```

```c
void process_arg(char *arg)

{
  char ch20;

  ch20 = arg[20];
  if (ch20 == '^') {
    secret_wisdom();
    return;
  }
  if (ch20 < '_') {
    if (ch20 == '#') {
      process_pound(arg);
      return;
    }
    if (ch20 == 'G') {
      process_G(arg);
      return;
    }
  }
  printf(arg);
  putchar(L'\n');
  return;
```

(The function(s) called by main are further examined, revealing that passing a string with ‘^’ at arg[20] to Puzzle will result in secret message output.)

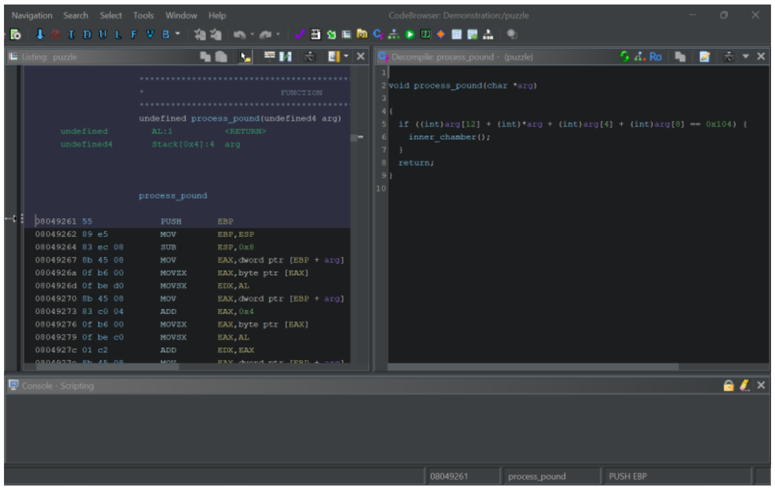
IV. Entering ‘puzzle 01234567890123456789^’ in Tiny Core Linux.



```
tc@box:~$ puzzle 01234567890123456789^
secret wisdom
tc@box:~$
```

(Puzzle prints ‘secret wisdom’ instead of the expected output.)

V. Further investigating Puzzle functions.

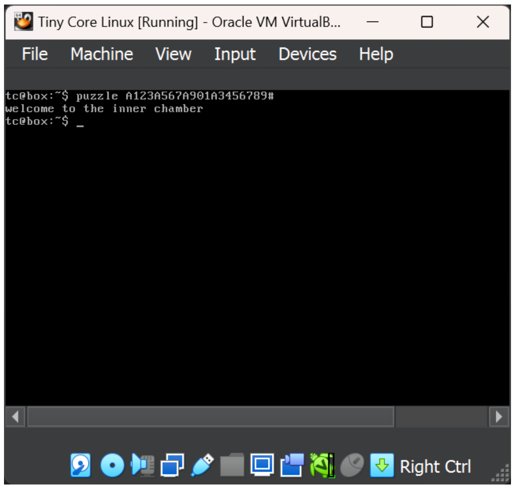


```
                  **************************************
                  *                FUNCTION
                  **************************************
                  undefined process_pound(undefined4 arg)
     undefined      AL:1          <RETURN>
     undefined4     Stack[0x4]:4  arg

                  process_pound
08049261 55              PUSH      EBP
08049262 89 e5           MOV       EBP,ESP
08049264 83 ec 08        SUB       ESP,0x8
08049267 8b 45 08        MOV       EAX,dword ptr [EBP + arg]
0804926a 0f b6 00        MOVZX     EAX,byte ptr [EAX]
0804926d 0f be d0        MOVSX     EDX,AL
08049270 8b 45 08        MOV       EAX,dword ptr [EBP + arg]
08049273 83 c0 04        ADD       EAX,0x4
08049276 0f b6 00        MOVZX     EAX,byte ptr [EAX]
08049279 0f be c0        MOVSX     EAX,AL
0804927c 01 c2           ADD       EDX,EAX
```

```c

void process_pound(char *arg)

{
  if ((int)arg[12] + (int)*arg + (int)arg[4] + (int)arg[8] == 0x104) {
    inner_chamber();
  }
  return;
}
```

(Upon further investigation, a function reveals that passing ‘#’ at arg[20] and ASCII values that add up to 260 at arg[0], arg[4], arg[8], and arg[12] to Puzzle will result in the output of a secret message.)

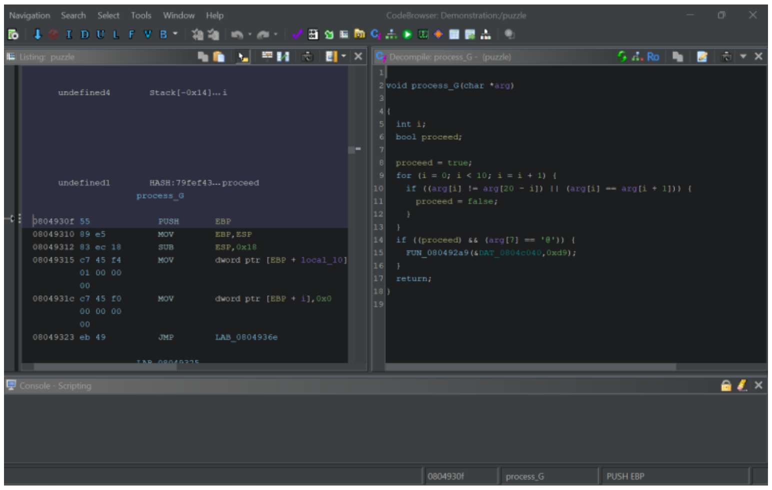
VI. Entering ‘puzzle A123A567A901A3456789#’ in Tiny Core Linux.



```
tc@box:~$ puzzle A123A567A901A3456789#
welcome to the inner chamber
tc@box:~$ _
```

(Puzzle prints ‘welcome to the inner chamber’ instead of the expected output.)

VII. Further investigating Puzzle functions.



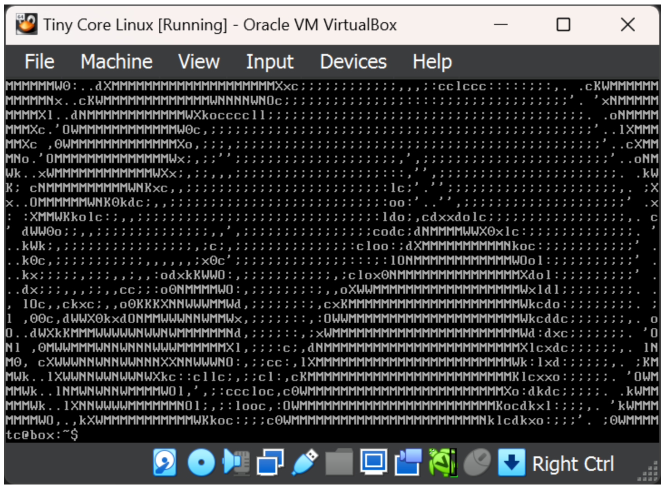
```
        undefined4        Stack[-0x14]...i

        undefined1        HASH:79fef43...proceed
                       process_G
0804930f 55              PUSH       EBP
08049310 89 e5           MOV        EBP,ESP
08049312 83 ec 18        SUB        ESP,0x18
08049315 c7 45 f4        MOV        dword ptr [EBP + local_10]
         01 00 00
         00
0804931c c7 45 f0        MOV        dword ptr [EBP + i],0x0
         00 00 00
         00
08049323 eb 49           JMP        LAB_0804936e
```

```
void process_G(char *arg)

{
  int i;
  bool proceed;

  proceed = true;
  for (i = 0; i < 10; i = i + 1) {
    if ((arg[i] != arg[20 - i]) || (arg[i] == arg[i + 1])) {
      proceed = false;
    }
  }
  if ((proceed) && (arg[7] == '@')) {
    FUN_080492a9(&DAT_0804c040,0xd9);
  }
  return;
}
```

```
0804930f    process_G    PUSH EBP
```

(Upon further investigation, a function reveals that passing ‘G’ at arg[20] and ‘@’ at arg[7] to Puzzle and ensuring the rest of the argument is a palindrome without repeating characters will result in the output of a secret message.)

VIII. Entering ‘puzzle GBABABA@ABABA@ABABABG’ in Tiny Core Linux.

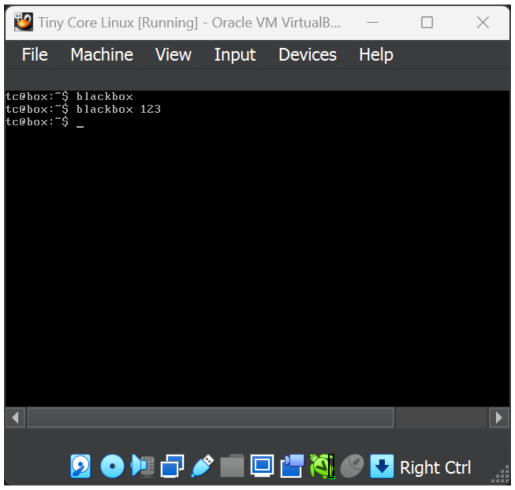


```
MMMMMMW0:..dXMMMMMMMMMMMMMMMMMMMMMMXxc;;;;;;;;;;;;,,,;:cclccc::::;;:,. .cKWMMMMMM
MMMMMNx..cKWMMMMMMMMMMMMMMWNNNNWN0c;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;'. 'xNMMMMM
MMMMXl..dNMMMMMMMMMMMMMMWXkocccccll;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;. .oNMMMM
MMMXc.'OWMMMMMMMMMMMMMW0c,;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;'..lXMMM
MMXc ,0WMMMMMMMMMMMMMMXo,;;;,;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;'..cXMM
MNo.'OMMMMMMMMMMMMMMMWx;,;;'';;;;;;;;;;;;;;;;;;;;',;;;;;;;;;;;;;;;;;;;;;;;;;;'..oNM
Wk..xWMMMMMMMMMMMMMWXx;,;;,,,;;;;;;;;;;;;;;;;;;;;;,'',;;;;;;;;;;;;;;;;;;;;;;;;;;. .kW
K; cNMMMMMMMMMMMMWNKxc,,;;;;;;;;;;;;;;;;;;;;;;;;;;;lc;'.'';;;;;;;;;;;;;;;;;;;;;;;,. ;X
x..OMMMMMMWNK0kdc;,,;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;oo:'..'',;;;;;;;;;;;;;;;;;;;;;;;' .x
: :XMMWKkolc:;,,;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;ldo;,cdxxdolc;;;;;;;;;;;;;;;;;;;;,. c
' dWW0o;;,,;;;;;;;;;;;;;;;;,,',;;;;;;;;;;;;;;;;;;;;codc;dNMMMMWWX0xlc:;;;;;;;;;;;;;;;;;. '
..kWk;,;;;;;;;;;;;;;;;;;;;,;c;,;;;;;;;;;;;;;;;;;;;:cloo:;dXMMMMMMMMMMMNkoc:;;;;;;;;;;;;;' .
..k0c,;;;;;;;;;;;;;,,,,,,;x0c';;;;;;;;;;;;;;;;;;;;;:lONMMMMMMMMMMMMMW0ol:;;;;;;;;;;;' .
..kx;;;;;,;;;,,;,,:odxkKWW0:,;;;;;;;;;;;;;;,;clox0NMMMMMMMMMMMMMMMMXdol:;;;;;;;;;;' .
..dx;;;,,,;;,,cc;;:o0NMMMMW0:,;;;;;;;;;;;,,oXWWMMMMMMMMMMMMMMMMMMMMMWxldl;;;;;;;;;;. .
, lOc,,ckxc;,,o0KKKXNNWWWMMWd,;;;;;;;;,cxKMMMMMMMMMMMMMMMMMMMMMMMMWkcdo;;;;;;;;;. ;
l ,00c,dWWX0kxdONMMWWWNNWMMWx,;;;;;;;,:OWWMMMMMMMMMMMMMMMMMMMMMMMMMWkcddc;;;;;;;,. o
O..dWXkKMMMWWWWWNWWNWMMMMMMMNd,;;;;;:,;xWMMMMMMMMMMMMMMMMMMMMMMMMMWd:dxc;;;;;;;;. 'O
Nl ,0MWWMMMWNNWNNNWWWMMMMMMXl,;;;:c;,dNMMMMMMMMMMMMMMMMMMMMMMMMMMMXlcxdc;;;;;;,. lN
M0, cXWWWNNWNNWWNNNXXNNWWWN0:,;;cc:,lXMMMMMMMMMMMMMMMMMMMMMMMMMWk:lxd:;;;;;,. ;KM
MWk..lXWWNNWWNWWNWXkc::cllc;,;;cl:,cKMMMMMMMMMMMMMMMMMMMMMMMMMKlcxxo:;;;;;. 'OWM
MMWk..lNMWNWNNWMMMMW0l,',;:cccloc,c0WMMMMMMMMMMMMMMMMMMMMMMMMMXo:dkdc;;;;;. .kWMM
MMMWk..lXNNWWWWMMMMMMN0l;,;:looc,:0WMMMMMMMMMMMMMMMMMMMMMMMMMKocdkxl:;;;,. 'kWMMM
MMMMW0,.,kXWMMMMMMMMMMMMWKkoc:;;;c0WMMMMMMMMMMMMMMMMMMMMMMMMMNklcdkxo:;;;'. ;0WMMMM
tc@box:~$
```

(Puzzle prints ASCII art of the Ghidra logo instead of the expected output.)

1.3 Black Box

I. Entering ‘blackbox’ and ‘blackbox 123’ in Tiny Core Linux.

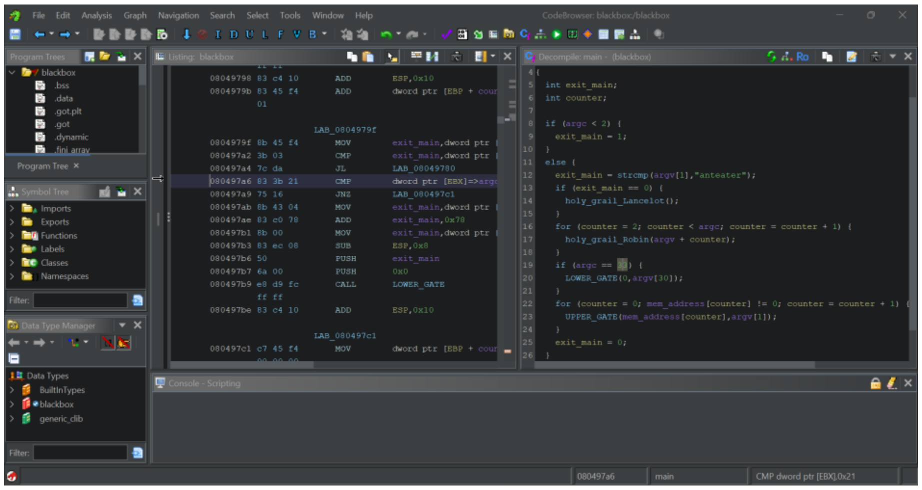


```
tc@box:~$ blackbox
tc@box:~$ blackbox 123
tc@box:~$ _
```

(Blackbox’s expected output when run directly or with args is nothing.)

GHIDRA SRE FINAL CHALLENGE

II. Investigating Blackbox's deconstructed contents and relabeling content in Ghidra



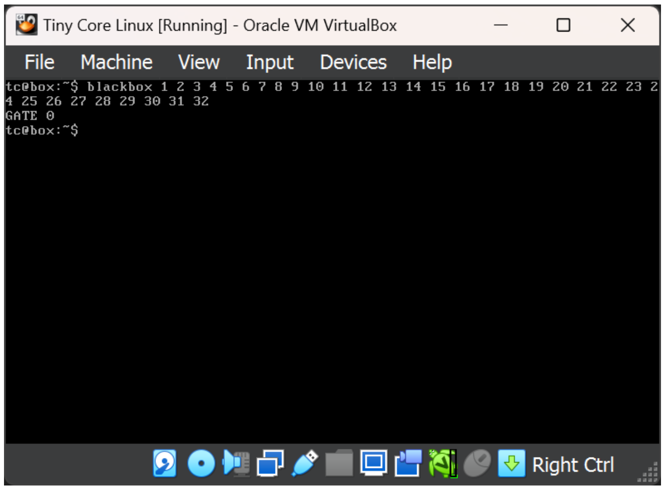
```
08049798 83 c4 10       ADD     ESP,0x10
0804979b 83 45 f4       ADD     dword ptr [EBP + cour
         01

                  LAB_0804979f
0804979f 8b 45 f4       MOV     exit_main,dword ptr
080497a2 3b 03          CMP     exit_main,dword ptr
080497a4 7c da          JL      LAB_08049780
080497a6 83 3b 21       CMP     dword ptr [EBX]=>argc
080497a9 75 16          JNZ     LAB_080497c1
080497ab 8b 43 04       MOV     exit_main,dword ptr
080497ae 83 c0 78       ADD     exit_main,0x78
080497b1 8b 00          MOV     exit_main,dword ptr
080497b3 83 ec 08       SUB     ESP,0x8
080497b6 50             PUSH    exit_main
080497b7 6a 00          PUSH    0x0
080497b9 e8 d9 fc       CALL    LOWER_GATE
         ff ff
080497be 83 c4 10       ADD     ESP,0x10

                  LAB_080497c1
080497c1 c7 45 f4       MOV     dword ptr [EBP + cour
```

```c
{
  int exit_main;
  int counter;

  if (argc < 2) {
    exit_main = 1;
  }
  else {
    exit_main = strcmp(argv[1],"anteater");
    if (exit_main == 0) {
      holy_grail_Lancelot();
    }
    for (counter = 2; counter < argc; counter = counter + 1) {
      holy_grail_Robin(argv + counter);
    }
    if (argc == 33) {
      LOWER_GATE(0,argv[30]);
    }
    for (counter = 0; mem_address[counter] != 0; counter = counter + 1) {
      UPPER_GATE(mem_address[counter],argv[1]);
    }
    exit_main = 0;
  }
```

080497a6 | main | CMP dword ptr [EBX],0x21

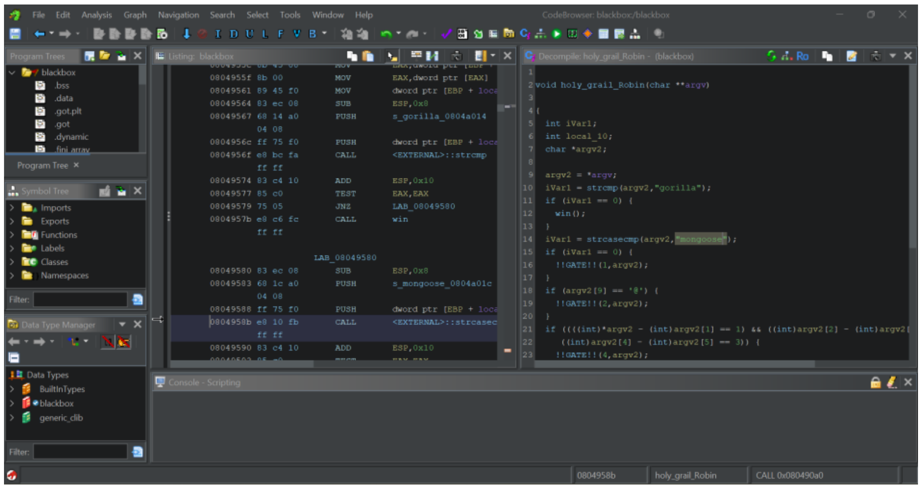
(Upon further investigation, an if check in a function will trigger gate 0 if 33 args are passed to blackbox [the function name is arg 1].)

III. Entering 'blackbox 1 2 3 … 30 31 32' in Tiny Core Linux.



```
tc@box:~$ blackbox 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 2
4 25 26 27 28 29 30 31 32
GATE 0
tc@box:~$
```

IV. Investigating Blackbox's deconstructed contents and relabeling content in Ghidra

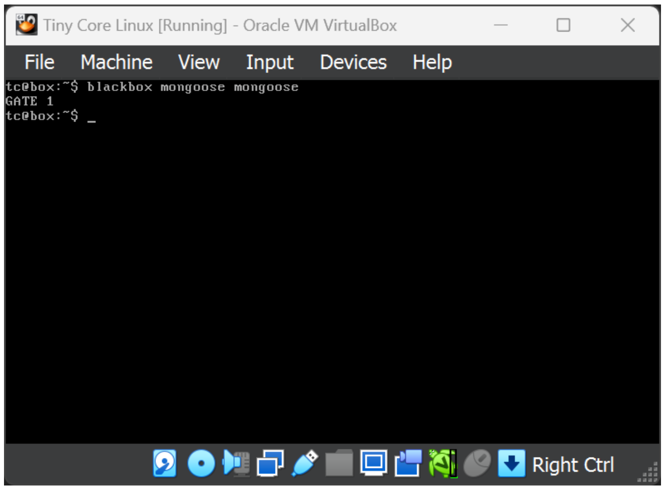


```
0804955f 8b 00          MOV     EAX,dword ptr [EAX]
08049561 89 45 f0       MOV     dword ptr [EBP + loca
08049564 83 ec 08       SUB     ESP,0x8
08049567 68 14 a0       PUSH    s_gorilla_0804a014
         04 08
0804956c ff 75 f0       PUSH    dword ptr [EBP + loca
0804956f e8 bc fa       CALL    <EXTERNAL>::strcmp
         ff ff
08049574 83 c4 10       ADD     ESP,0x10
08049577 85 c0          TEST    EAX,EAX
08049579 75 05          JNZ     LAB_08049580
0804957b e8 c6 fc       CALL    win
         ff ff

                   LAB_08049580
08049580 83 ec 08       SUB     ESP,0x8
08049583 68 1c a0       PUSH    s_mongoose_0804a01c
         04 08
08049588 ff 75 f0       PUSH    dword ptr [EBP + loca
0804958b e8 10 fb       CALL    <EXTERNAL>::strcasec
         ff ff
08049590 83 c4 10       ADD     ESP,0x10
```

```
void holy_grail_Robin(char **argv)

{
  int iVar1;
  int local_10;
  char *argv2;

  argv2 = *argv;
  iVar1 = strcmp(argv2,"gorilla");
  if (iVar1 == 0) {
    win();
  }
  iVar1 = strcasecmp(argv2,"mongoose");
  if (iVar1 == 0) {
    !!GATE!!(1,argv2);
  }
  if (argv2[9] == '@') {
    !!GATE!!(2,argv2);
  }
  if ((((int)*argv2 - (int)argv2[1] == 1) && ((int)argv2[2] - (int)argv2[
     ((int)argv2[4] - (int)argv2[5] == 3)) {
    !!GATE!!(4,argv2);
```

(Upon further investigation, an if check in a function will trigger gate 1 if the second arg passed to blackbox is 'mongoose'.)

V. Entering 'blackbox mongoose mongoose' in Tiny Core Linux.

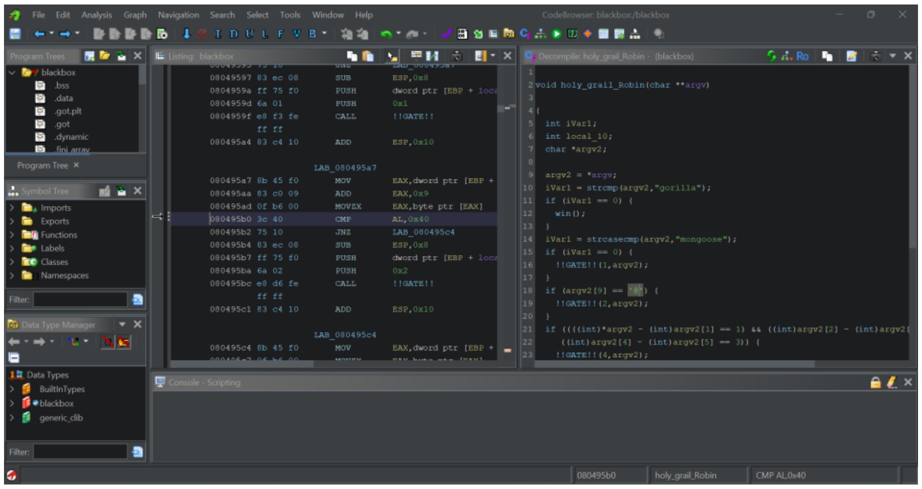


```
tc@box:~$ blackbox mongoose mongoose
GATE 1
tc@box:~$ _
```

(Blackbox prints GATE 1 instead of the expected output.)

GHIDRA SRE FINAL CHALLENGE

VI. Investigating Blackbox's deconstructed contents and relabeling content in Ghidra.

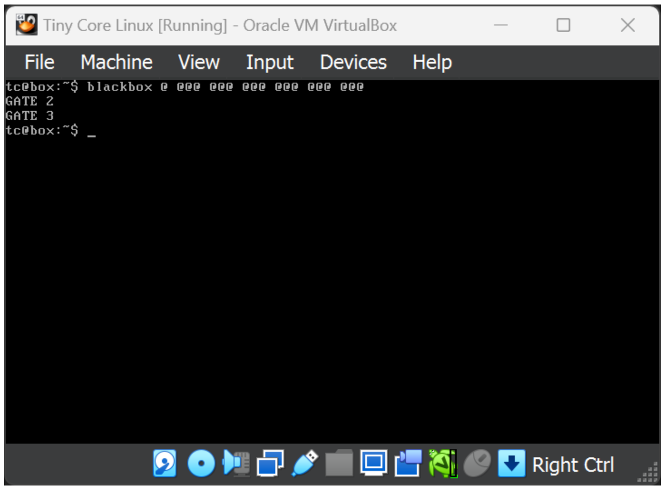


```
08049597 83 ec 08        SUB      ESP,0x8
0804959a ff 75 f0        PUSH     dword ptr [EBP + loca
0804959d 6a 01           PUSH     0x1
0804959f e8 f3 fe        CALL     !!GATE!!
         ff ff
080495a4 83 c4 10        ADD      ESP,0x10

                    LAB_080495a7
080495a7 8b 45 f0        MOV      EAX,dword ptr [EBP +
080495aa 83 c0 09        ADD      EAX,0x9
080495ad 0f b6 00        MOVZX    EAX,byte ptr [EAX]
080495b0 3c 40           CMP      AL,0x40
080495b2 75 10           JNZ      LAB_080495c4
080495b4 83 ec 08        SUB      ESP,0x8
080495b7 ff 75 f0        PUSH     dword ptr [EBP + loca
080495ba 6a 02           PUSH     0x2
080495bc e8 d6 fe        CALL     !!GATE!!
         ff ff
080495c1 83 c4 10        ADD      ESP,0x10

                    LAB_080495c4
080495c4 8b 45 f0        MOV      EAX,dword ptr [EBP +
```

```c

void holy_grail_Robin(char **argv)

{
  int iVar1;
  int local_10;
  char *argv2;

  argv2 = *argv;
  iVar1 = strcmp(argv2,"gorilla");
  if (iVar1 == 0) {
    win();
  }
  iVar1 = strcasecmp(argv2,"mongoose");
  if (iVar1 == 0) {
    !!GATE!!(1,argv2);
  }
  if (argv2[9] == '@') {
    !!GATE!!(2,argv2);
  }
  if ((((int)*argv2 - (int)argv2[1] == 1) && ((int)argv2[2] - (int)argv2[
     ((int)argv2[4] - (int)argv2[5] == 3)) {
    !!GATE!!(4,argv2);
```

(Upon further investigation, an if check in a function will trigger gate 2 if the second arg passed to blackbox has '@' at position 10.)

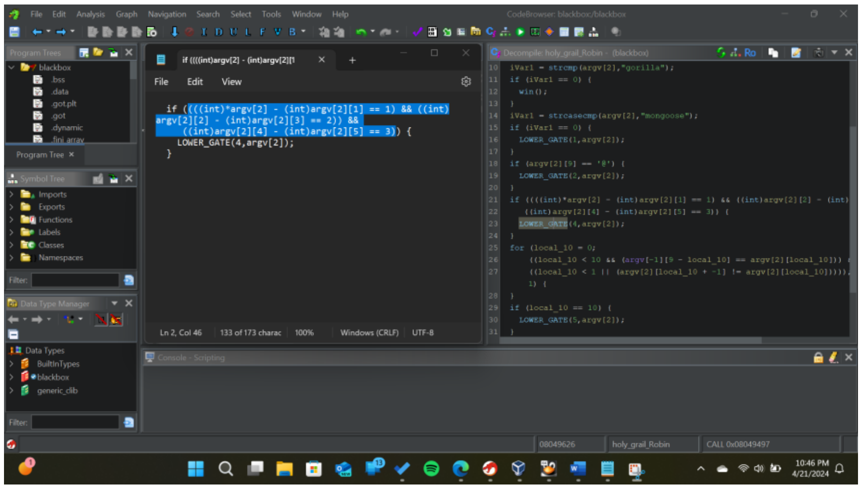
VII. Entering 'blackbox @ @@@ @@@ @@@ @@@ @@@" in Tiny Core Linux.



```
tc@box:~$ blackbox @ @@@ @@@ @@@ @@@ @@@ @@@
GATE 2
GATE 3
tc@box:~$ _
```

(Blackbox prints GATE 2 [and GATE 3 too, inadvertently] instead of the expected output.)

VIII. Investigating Blackbox's deconstructed contents and relabeling content in Ghidra.



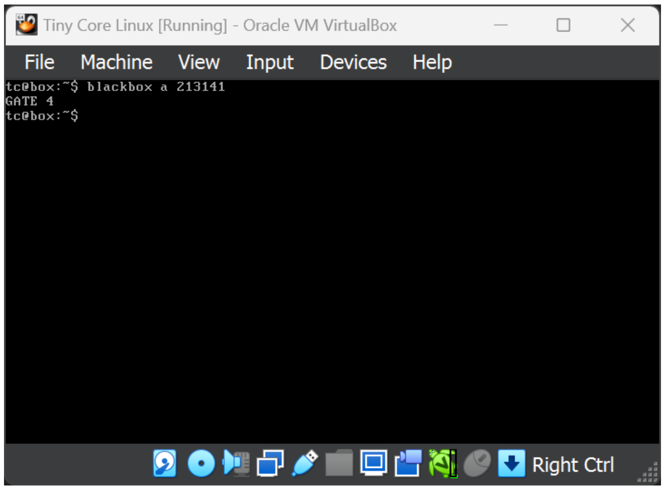
```
  if ((((int)*argv[2] - (int)argv[2][1] == 1) && ((int)
argv[2][2] - (int)argv[2][3] == 2)) &&
      ((int)argv[2][4] - (int)argv[2][5] == 3)) {
    LOWER_GATE(4,argv[2]);
  }
```

Decompile: holy_grail_Robin - (blackbox)

```
  iVar1 = strcmp(argv[2],"gorilla");
  if (iVar1 == 0) {
    win();
  }
  iVar1 = strcasecmp(argv[2],"mongoose");
  if (iVar1 == 0) {
    LOWER_GATE(1,argv[2]);
  }
  if (argv[2][9] == '@') {
    LOWER_GATE(2,argv[2]);
  }
  if ((((int)*argv[2] - (int)argv[2][1] == 1) && ((int)argv[2][2] - (int)
     ((int)argv[2][4] - (int)argv[2][5] == 3)) {
    LOWER_GATE(4,argv[2]);
  }
  for (local_10 = 0;
      ((local_10 < 10 && (argv[-1][9 - local_10] == argv[2][local_10])) 
      ((local_10 < 1 || (argv[2][local_10 + -1] != argv[2][local_10]))));
      1) {
  }
  if (local_10 == 10) {
    LOWER_GATE(5,argv[2]);
  }
```

(Upon further investigation, an if check in a function will trigger gate 4 if the second arg passed to blackbox adheres to this pattern: [argv[2][0] – argv[2][1] = 1, argv[2][2] - argv[2][3] = 2, and argv[2][4] - argv[2][5] = 3.)

IX. Entering 'blackbox a 213141' in Tiny Core Linux.

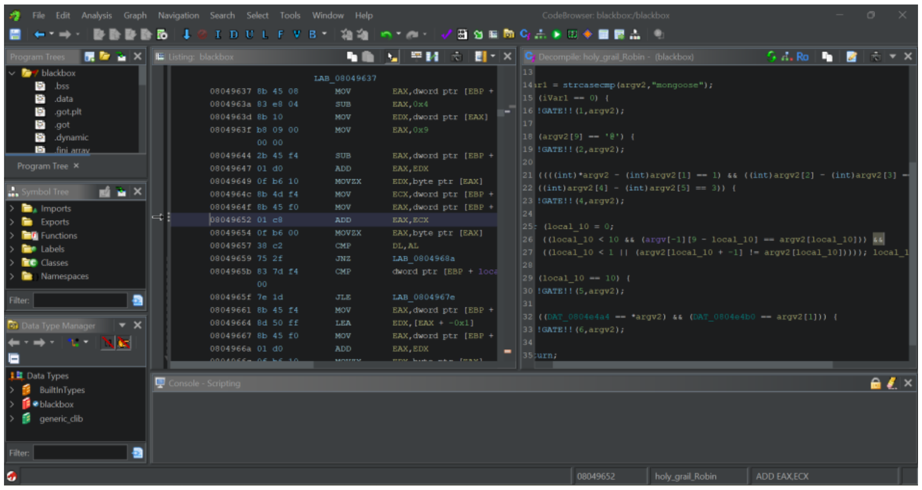


```
tc@box:~$ blackbox a 213141
GATE 4
tc@box:~$
```

(Blackbox prints GATE 4 instead of the expected output [2 – 1 = 1, 3 – 1 = 2, and 4 – 1 = 3].)

GHIDRA SRE FINAL CHALLENGE

X. Investigating Blackbox's deconstructed contents and relabeling content in Ghidra.

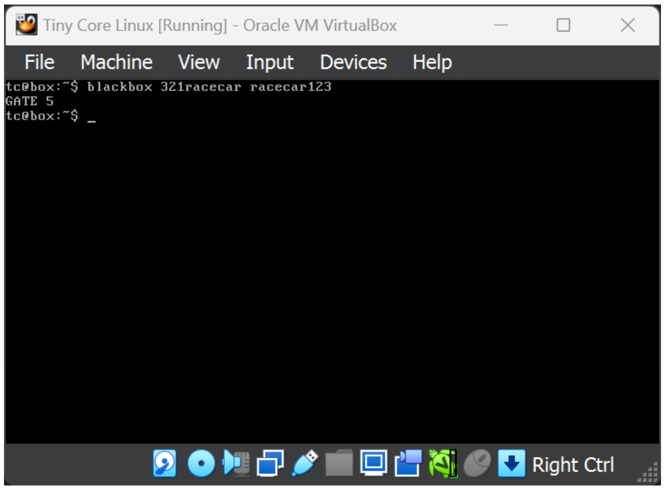


```
                         LAB_08049637
        08049637 8b 45 08       MOV      EAX,dword ptr [EBP +
        0804963a 83 e8 04       SUB      EAX,0x4
        0804963d 8b 10          MOV      EDX,dword ptr [EAX]
        0804963f b8 09 00       MOV      EAX,0x9
                 00 00
        08049644 2b 45 f4       SUB      EAX,dword ptr [EBP +
        08049647 01 d0          ADD      EAX,EDX
        08049649 0f b6 10       MOVZX    EDX,byte ptr [EAX]
        0804964c 8b 4d f4       MOV      ECX,dword ptr [EBP +
        0804964f 8b 45 f0       MOV      EAX,dword ptr [EBP +
        08049652 01 c8          ADD      EAX,ECX
        08049654 0f b6 00       MOVZX    EAX,byte ptr [EAX]
        08049657 38 c2          CMP      DL,AL
        08049659 75 2f          JNZ      LAB_0804968a
        0804965b 83 7d f4       CMP      dword ptr [EBP + loca
                 00
        0804965f 7e 1d          JLE      LAB_0804967e
        08049661 8b 45 f4       MOV      EAX,dword ptr [EBP +
        08049664 8d 50 ff       LEA      EDX,[EAX + -0x1]
        08049667 8b 45 f0       MOV      EAX,dword ptr [EBP +
        0804966a 01 d0          ADD      EAX,EDX
```

```

ar1 = strcasecmp(argv2,"mongoose");
 (iVar1 == 0) {
 !GATE!!(1,argv2);

 (argv2[9] == '@') {
 !GATE!!(2,argv2);

 ((((int)*argv2 - (int)argv2[1] == 1) && ((int)argv2[2] - (int)argv2[3] -
 ((int)argv2[4] - (int)argv2[5] == 3)) {
 !GATE!!(4,argv2);

  (local_10 = 0;
  ((local_10 < 10 && (argv[-1][9 - local_10] == argv2[local_10])) &&
  ((local_10 < 1 || (argv2[local_10 + -1] != argv2[local_10])))); local_1

 (local_10 == 10) {
 !GATE!!(5,argv2);

 ((DAT_0804e4a4 == *argv2) && (DAT_0804e4b0 == argv2[1])) {
 !GATE!!(6,argv2);

urn;
```

(Upon further investigation, an if check in a function will trigger gate 5 if the args passed to the argv[-1], the preceding arg, match arg[2], a palindrome, in reverse when passed to blackbox.)

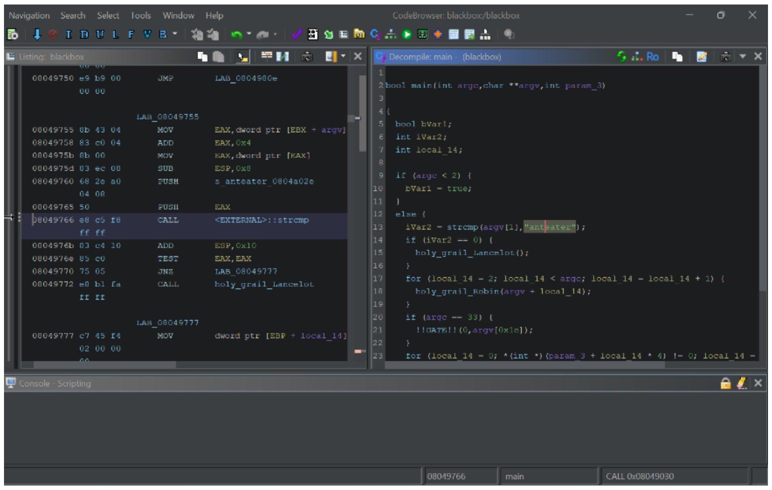
XI. Entering 'blackbox 321racecar racecar123' in Tiny Core Linux.



```
tc@box:~$ blackbox 321racecar racecar123
GATE 5
tc@box:~$ _
```

(Blackbox prints GATE 5 instead of the expected output.)

XII. Investigating Blackbox’s deconstructed contents and relabeling content in Ghidra.


Navigation Search Select Tools Window Help
CodeBrowser: blackbox/blackbox
Listing: blackbox
Decompile: main - (blackbox)
Console - Scripting

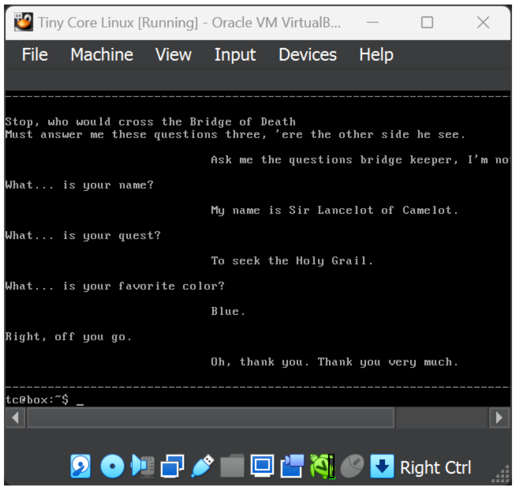
```
08049750 e9 b9 00       JMP        LAB_0804980e
         00 00

                   LAB_08049755
08049755 8b 43 04       MOV        EAX,dword ptr [EBX + argv]
08049758 83 c0 04       ADD        EAX,0x4
0804975b 8b 00          MOV        EAX,dword ptr [EAX]
0804975d 83 ec 08       SUB        ESP,0x8
08049760 68 2e a0       PUSH       s_anteater_0804a02e
         04 08
08049765 50             PUSH       EAX
08049766 e8 c5 f8       CALL       <EXTERNAL>::strcmp
         ff ff
0804976b 83 c4 10       ADD        ESP,0x10
0804976e 85 c0          TEST       EAX,EAX
08049770 75 05          JNZ        LAB_08049777
08049772 e8 b1 fa       CALL       holy_grail_Lancelot
         ff ff

                   LAB_08049777
08049777 c7 45 f4       MOV        dword ptr [EBP + local_14]
         02 00 00
```

```
bool main(int argc,char **argv,int param_3)

{
  bool bVar1;
  int iVar2;
  int local_14;

  if (argc < 2) {
    bVar1 = true;
  }
  else {
    iVar2 = strcmp(argv[1],"anteater");
    if (iVar2 == 0) {
      holy_grail_Lancelot();
    }
    for (local_14 = 2; local_14 < argc; local_14 = local_14 + 1) {
      holy_grail_Robin(argv + local_14);
    }
    if (argc == 33) {
      !!GATE!!(0,argv[0x1e]);
    }
    for (local_14 = 0; *(int *)(param_3 + local_14 * 4) != 0; local_14 =
```

```
08049766    main    CALL 0x08049030
```

(Upon further examination of the main entry point of the program’s comparisons, it’s revealed that passing a string with ‘anteater’ at arg[1] to Blackbox will result in secret message output.)

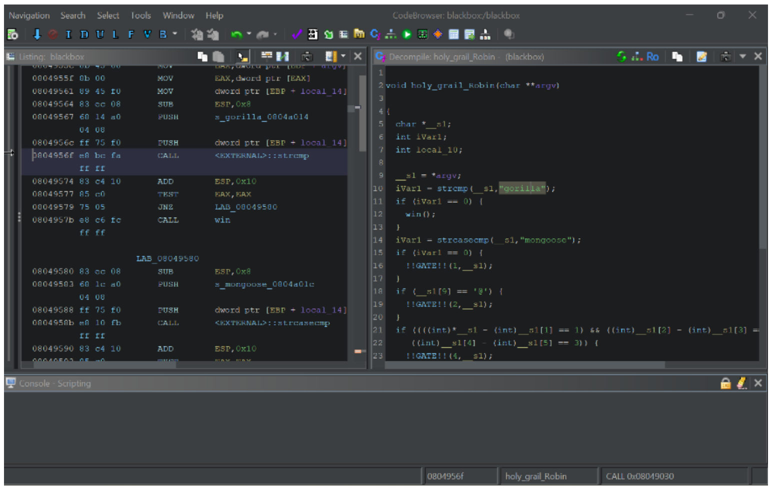
XIII. Entering ‘blackbox anteater’ in Tiny Core Linux.


Tiny Core Linux [Running] - Oracle VM VirtualB...
File Machine View Input Devices Help
Right Ctrl

```
------------------------------------------------------------------------
Stop, who would cross the Bridge of Death
Must answer me these questions three, 'ere the other side he see.
                            Ask me the questions bridge keeper, I'm no
What... is your name?
                            My name is Sir Lancelot of Camelot.
What... is your quest?
                            To seek the Holy Grail.
What... is your favorite color?
                            Blue.
Right, off you go.
                            Oh, thank you. Thank you very much.
------------------------------------------------------------------------
tc@box:~$ _
```

(Blackbox prints a Monty Python dialogue instead of the expected output.)

XIV. Further investigating Blackbox functions.

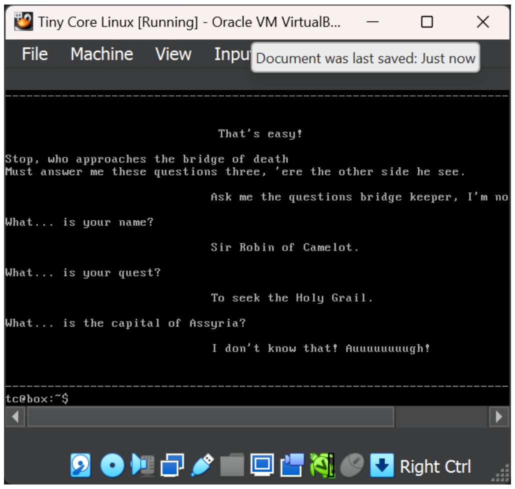

CodeBrowser: blackbox/blackbox
Navigation Search Select Tools Window Help
Listing: blackbox
Decompile: holy_grail_Robin - (blackbox)
Console - Scripting
0804956f
holy_grail_Robin
CALL 0x08049030

```
0804955f 8b 00        MOV    EAX,dword ptr [EAX]
08049561 89 45 f0     MOV    dword ptr [EBP + local_14]
08049564 83 ec 08     SUB    ESP,0x8
08049567 68 14 a0     PUSH   s_gorilla_0804a014
         04 08
0804956c ff 75 f0     PUSH   dword ptr [EBP + local_14]
0804956f e8 bc fa     CALL   <EXTERNAL>::strcmp
         ff ff
08049574 83 c4 10     ADD    ESP,0x10
08049577 85 c0        TEST   EAX,EAX
08049579 75 05        JNZ    LAB_08049580
0804957b e8 c6 fc     CALL   win
         ff ff

                 LAB_08049580
08049580 83 ec 08     SUB    ESP,0x8
08049583 68 1c a0     PUSH   s_mongoose_0804a01c
         04 08
08049588 ff 75 f0     PUSH   dword ptr [EBP + local_14]
0804958b e8 10 fb     CALL   <EXTERNAL>::strcasecmp
         ff ff
08049590 83 c4 10     ADD    ESP,0x10
```

```c

void holy_grail_Robin(char **argv)

{
  char *__s1;
  int iVar1;
  int local_10;

  __s1 = *argv;
  iVar1 = strcmp(__s1,"gorilla");
  if (iVar1 == 0) {
    win();
  }
  iVar1 = strcasecmp(__s1,"mongoose");
  if (iVar1 == 0) {
    !!GATE!!(1,__s1);
  }
  if (__s1[9] == '@') {
    !!GATE!!(2,__s1);
  }
  if ((((int)*__s1 - (int)__s1[1] == 1) && ((int)__s1[2] - (int)__s1[3] =
     ((int)__s1[4] - (int)__s1[5] == 3)) {
    !!GATE!!(4,__s1);
```

(Upon further investigation, a function reveals that if arg[1] and arg[2] match as 'gorilla' when passed to Blackbox, then it will result in the output of a secret message.)

XV. Entering 'blackbox gorilla gorilla' in Tiny Core Linux.


Tiny Core Linux [Running] - Oracle VM VirtualB...
File Machine View Inpu
Document was last saved: Just now
Right Ctrl

```
------------------------------------------------------------------------

                              That's easy!

Stop, who approaches the bridge of death
Must answer me these questions three, 'ere the other side he see.

                             Ask me the questions bridge keeper, I'm no

What... is your name?

                             Sir Robin of Camelot.

What... is your quest?

                             To seek the Holy Grail.

What... is the capital of Assyria?

                             I don't know that! Auuuuuuuugh!

------------------------------------------------------------------------
tc@box:~$
```

(Blackbox prints another Monty Python dialogue instead of the expected output.)

1.4 Conclusion

I’ve successfully enhanced my reverse engineering skills by using Ghidra and VirtualBox to analyze and decompile binaries (‘Puzzle’ and ‘Black Box’), as well as identifying and manipulating functions within the ‘Puzzle’ and ‘Black Box’ binaries to reveal secret messages and behaviors triggered by specific string conditions. This hands-on experience improved my problem-solving capabilities and deepened my understanding of binary functions and conditional logic. Through my completion of these activities, to the best of my ability, I’ve gained proficiency in using advanced tools and techniques to explore and manipulate software at a low level.