
Lab 3: Binary Bomb Lab

KAIST CS230 Fall 2021

Due date: Tue 12th, Oct 2021

- The ‘bomb’ file is a binary executable file.
 - Bomb includes 6 stages to defuse + 1 hidden stage
 - Only main routine will be given as source code.
-

Preparing Workspace



- Get bomb at <http://143.248.188.188:15213/>
 - It will be downloaded as .tar file
 - Move .tar file to your linux server
 - You will need to use scp tools
 - Decompress .tar file by typing 'tar -xvf <file name>'
 - Now, deal with bomb under generated directory!
-

CS:APP Binary Bomb Request

Fill in the form and then click the Submit button.

Hit the Reset button to get a clean form.

Legal characters are spaces, letters, numbers, underscores ('_'), hyphens ('-'), at signs ('@'), and dots ('.').

Student ID

(ex: 20190000)

KAIST e-mail address

Submit

Reset

- Unlike previous labs, we're not using gitlab.
 - Check your score at
<http://143.248.188.188:15213/scoreboard>
 - This time, you don't need to submit your solution.
 - Just defuse your bomb at provided linux server!
-

Bomb Lab Scoreboard

This page contains the latest information that we have received from your bomb. If your solution is marked **invalid**, this means your bomb reported a solution that didn't actually defuse your bomb.

Last updated: Mon Sep 23 20:47:34 2019 (updated every 30 secs)

#	Bomb number	Submission date	Phases defused	Explosions	Score	Status
1	bomb1	Mon Sep 23 20:47	2	2	19	valid

Summary [phase:cnt] [1:0] [2:1] [3:0] [4:0] [5:0] [6:0] total defused = 0/1

- GNU Debugger
- Shows machine state in real time

```
~ ➔ gdb
GNU gdb (GDB) 8.2
Copyright (C) 2018 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-apple-darwin17.7.0".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
  <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) █
```

- Start GDB: `gdb [options] [executable name]`

```
ta@canis01:~$ gdb bomb
GNU gdb (Ubuntu 7.7.1-0ubuntu5~14.04.3) 7.7.1
Copyright (C) 2014 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-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from bomb...done.
(gdb) █
```

- Show functions: i(nfo) f(unc)

```
~ ➔ cat test.c
#include <stdio.h>

int add(int a, int b){
    return a+b;
}

int sub(int a, int b){
    return a-b;
}

int main(int argc, char** argv){
    int i=3;
    int j=2;

    i=add(i, j);
    j=sub(i, j);

}
```

```
(gdb) info func
All defined functions:

Non-debugging symbols:
0x0000000100000000  _mh_execute_header
0x0000000100000f30  add
0x0000000100000f50  sub
0x0000000100000f70  main
(gdb) █
```

- View assembly of a function:
disas(semble) [function name]

```
~ ➔ cat test.c
#include <stdio.h>

int add(int a, int b){
    return a+b;
}

int sub(int a, int b){
    return a-b;
}

int main(int argc, char** argv){
    int i=3;
    int j=2;

    i=add(i, j);
    j=sub(i, j);

}
```

```
(gdb) disassemble main
Dump of assembler code for function main:
0x0000000004004fc <+0>:    push   %rbp
0x0000000004004fd <+1>:    mov    %rsp,%rbp
0x000000000400500 <+4>:    sub    $0x20,%rsp
0x000000000400504 <+8>:    mov    %edi,-0x14(%rbp)
0x000000000400507 <+11>:   mov    %rsi,-0x20(%rbp)
0x00000000040050b <+15>:   movl   $0x3,-0x8(%rbp)
0x000000000400512 <+22>:   movl   $0x2,-0x4(%rbp)
0x000000000400519 <+29>:   mov    -0x4(%rbp),%edx
0x00000000040051c <+32>:   mov    -0x8(%rbp),%eax
0x00000000040051f <+35>:   mov    %edx,%esi
0x000000000400521 <+37>:   mov    %eax,%edi
0x000000000400523 <+39>:   callq  0x4004d6 <add>
0x000000000400528 <+44>:   mov    %eax,-0x8(%rbp)
0x00000000040052b <+47>:   mov    -0x4(%rbp),%edx
0x00000000040052e <+50>:   mov    -0x8(%rbp),%eax
0x000000000400531 <+53>:   mov    %edx,%esi
0x000000000400533 <+55>:   mov    %eax,%edi
0x000000000400535 <+57>:   callq  0x4004ea <sub>
0x00000000040053a <+62>:   mov    %eax,-0x4(%rbp)
0x00000000040053d <+65>:   mov    $0x0,%eax
0x000000000400542 <+70>:   leaveq 
0x000000000400543 <+71>:   retq

End of assembler dump.
```

- Run program: r(un)

```
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 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-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from test...(no debugging symbols found)...done.
(gdb) run
Starting program: /home/elpis/test
[Inferior 1 (process 74355) exited normally]
(gdb) █
```

- Breakpoints: Makes program to stop at certain point
 - b(reak) [line number]
 - b(reak) [function name]
 - b(reak) [file name]:[line number]
 - b(reak) *[address]

```
(gdb) b main
Breakpoint 1 at 0x400500
(gdb) r
Starting program: /home/elpis/test

Breakpoint 1, 0x000000000400500 in main ()
(gdb) █
```

- List current breakpoints: i(nfo) b(reakpoint)
- Disable breakpoint: disable [breakpoint number]
- Enable breakpoint: enable [breakpoint number]
- Delete breakpoint: d(elete) [breakpoint number]
- Execute until breakpoint(or end of file): c(ontinue)
- Execute next instruction: n(ext)i

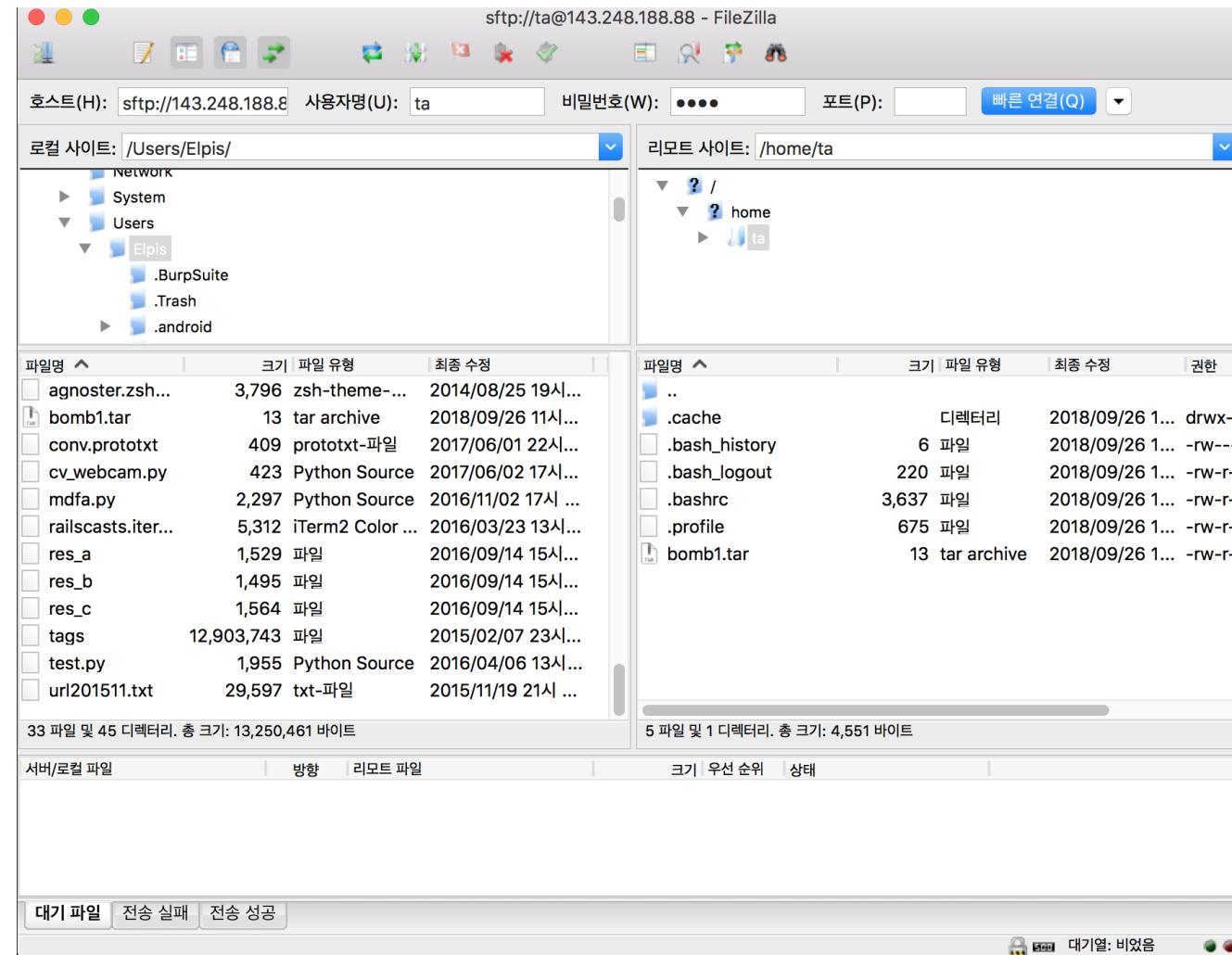
- More useful resources
 - <https://youtu.be/svG6OPyKsrw>
 - <https://youtu.be/sCtY--xRUyl>
-

- Copy a file from a machine to another
- WinSCP, FileZilla: for Windows (FileZilla is also for Mac/Linux)
 - User-friendly interface
- SCP: for Mac/Linux
 - built-in commands in the shell
- **Don't forget to turn on your KVPN**

Download links: FileZilla - <https://filezilla-project.org/>

WinSCP - <https://winscp.net/eng/download.php>

Example: FileZilla



- Execute as following on shell:
> scp [target file location] [username]@[server IP]:
[destination(directory)]
- Example: scp bomb1.tar ta@143.248.188.188:~
 - Copy **bomb1.tar** from local machine's current directory to home directory(~) of account **ta** in **143.248.188.188**

- You'd better **not work** with multiple bombs
 - Explosion signal from every bomb will be reported
 - You must download bomb **using your information**
 - Using other's information will be considered as cheating
 - You should run your binary bomb on **class machine**
 - If not, the bomb will refuse to run.
 - This is an **individual project**
-

- You may prevent explosion by using GDB breakpoint
 - Note that breakpoints are volatile
- Secret phase
 - For more challenge, you can find out secret phase.
- All bombs are different
 - So, do not try your friend's answer on your bomb ☺

- Permission denied error
 - Use chmod command to change permission
 - Cannot execute: chmod +x
 - Cannot read: chmod +r
 - Cannot write: chmod +w
- Invalid host error
 - Work on the class server
- SCP(FileZilla) connection failure
 - Check KVPN again