CENG331
Computer Organization
Bomb Lab Recitation

Fall 2021

Outline

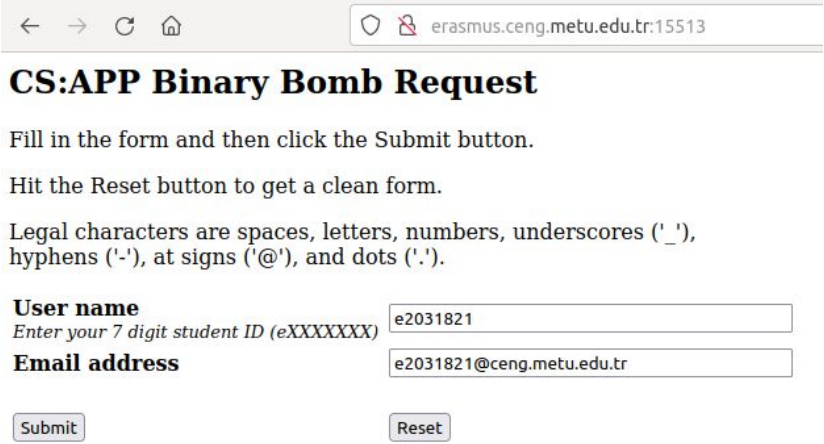
- Introduction
- Getting Started
- Useful Commands
- Tracing Assembly with gdb
- Resources, Tips

Introduction

- Goal: Deactivating “bombs”, compiled binaries that you need to enter specific strings to “defuse” phases.
- 6 phases in each bomb, each bomb has different set of phases and solution strings.
- practice your assembly reading skills and understand the way compiler converts C to x86.

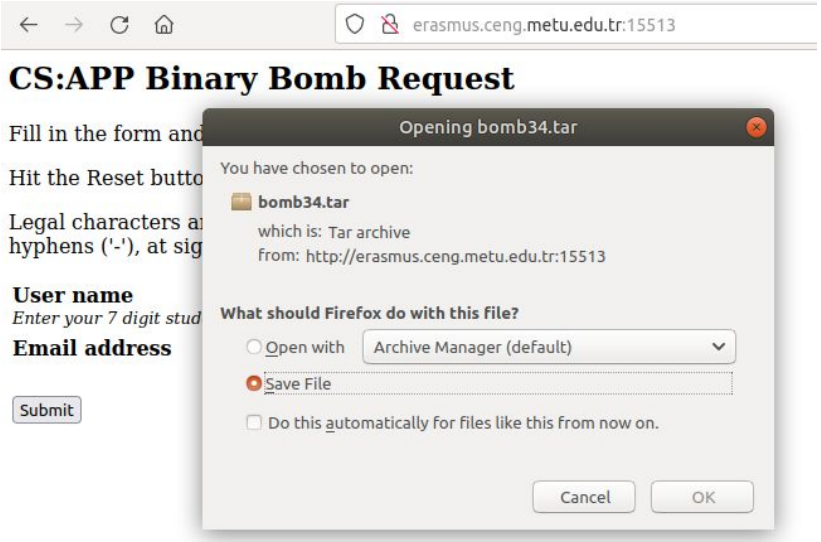
Getting started

- Get your bomb from http://erasmus.ceng.metu.edu.tr:15513
- you need to be within the campus network or use METUVPN to see this website



Getting started

- Get your bomb from erasmus.ceng.metu.edu.tr:15513
- you need to be within the campus network or use METUVPN to see this website



Getting started

- put the bomb to your department storage (through sftp/scp with external.ceng.metu.edu.tr or other means)
- run/debug it on ineks, you have to use one of inek[1,100] otherwise the bomb will not run. You can use ssh.

Getting started

- Your important actions (bomb defusals, explosions) will be notified to the bomb server. View them at http://erasmus.ceng.metu.edu.tr:15513/scoreboard

erasmus.ceng.metu.edu.tr:15513/scoreboard

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: Thu Nov 11 15:36:41 2021 (updated every 30 secs)

#	Bomb number	Submission date	Phases defused	Explosions	Score	Status
1	bomb6	Wed Nov 10 17:57	7	0	100	valid
2	bomb23	Wed Nov 10 16:14	5	0	76	valid
3	bomb10	Thu Nov 11 15:26	3	0	35	valid
4	bomb25	Wed Nov 10 22:41	2	3	19	valid
5	bomb24	Thu Nov 11 14:49	1	0	10	valid
6	bomb18	Tue Nov 9 20:21	1	1	10	valid
7	bomb1	Thu Nov 11 00:56	1	1	10	valid
8	bomb19	Tue Nov 9 20:17	1	8	6	valid

Summary [phase:cnt] [1:4] [2:1] [3:1] [4:0] [5:1] [6:0] [7:1] total defused = 1/8

Useful Commands

- objdump -d <objfile> Disassembles instruction related parts of the object file.
- strings <file> prints printable strings of length >= 4 found in the file.
- objdump -t <objfile> prints the symbol table of the object file.

Tracing Assembly with gdb

- > gdb bomb := start gdb with bomb
- gdb> run <cmd_args> := run program with cmd_args
- gdb> break <addr or label> := put a break point to the specified label or addr
- gdb> info break := list active breakpoints
- gdb> delete <#> := delete breakpoint with number “#”
- gdb> continue := run program until a breakpoint is hit
- gdb> stepi := run a single instruction
- gdb> nexti := run a single instruction, if it is a function call, run program until function returns
- gdb> kill := terminate the program
- gdb> disas := lists assembly code of the current function
- gdb> disas <addr/label/function> := lists assembly around instruction addr, label or for the whole function.

Tracing Assembly with gdb

gdb> print ($rsp) := print contents of %rsp as decimal signed number

gdb> print /x ($rsp) := print contents of %rsp as hex

gdb> print /u *(int *)($rsp+8) := print the int at the address “%rsp+8”. This essentially reads 4 bytes (int is 4 bytes) starting from the address “%rsp+8” as an unsigned decimal number.

gdb> print /s *(char *)(($rsp+2)+1)@10 := print 10 contiguous chars starting from one char away from adress “%rsp+2” as a c string. *Very useful!*

Tracing Assembly with gdb

- gdb> tui <enable/disable> := enables/disables a more gui like view
- gdb> layout <asm/regs/source> := changes tui view to your liking
- focus <cmd/asm/regs> focuses cursor for the specified window in tui mode.

Resources, Tips

- *Highly recommended read*: Chapter 3 of your textbook. You will be able to make accurate educated guesses with the knowledge and finish the assignment quicker. It will also help you skip over the unimportant sections in the code.
- Nice cheatsheet about gdb: http://csapp.cs.cmu.edu/3e/docs/gdbnotes-x86-64.pdf
- Your homework text has some resources and tips listed.