

[stackzero.net](https://www.stackzero.net)

Cracking PicoCTF Challenge: GDB Baby Step 1 - StackZero

stackzero

7-9 minutes

Are you a beginner in the intriguing world of reverse engineering? Or perhaps you're keen to delve into the workings of Linux's GDB debugger? Either way, we've got you covered. Welcome to [Stackzero](https://www.stackzero.net), your one-stop destination for all things related to these topics. Here, we've curated a wealth of resources to guide your learning journey.

If you're just starting out, we recommend our foundation guides on [Reverse Engineering](#), [GDB](#) and maybe the [Mastering PicoCTF: Your Ultimate Registration Guide!](#)

These easy-to-understand tutorials cover the basics of reverse engineering and using the GDB debugger. So why not take a few moments to read through them? It will give you the background knowledge you need to tackle the more complex challenges that lie ahead.

Today, we're going to take on a particular challenge – the PicoCTF “GDB Baby Step 1”.

It's an exciting task that tests your understanding of both reverse engineering and the GDB debugger.

Don't worry if you're new to these concepts, we'll walk through it step by step.

Ready to embark on this exciting journey of discovery? Let's jump in!

- [Decoding GDB: A Primer to the GNU Debugger](#)
- [Preparations and File Analysis](#)
- [The Magic of Disassembling with GDB](#)
- [Alternative Path: Debugging the Executable](#)
- [Conclusion](#)

Decoding [GDB](#): A Primer to the GNU Debugger

The world of programming and debugging can seem complex. Fortunately, the GNU Debugger, known as GDB, is here to help. GDB is a useful tool for programmers, especially those using Linux systems.

In essence, it lets you see what's happening inside your program while it's running. You can explore your code in depth, just like examining a machine's parts closely.

With GDB, you can disassemble code. This means breaking a program down into smaller parts, a handy feature when working with low-level languages or inspecting compiled programs.

GDB also lets you inspect registers – high-speed storage areas in a computer's CPU. Knowing what's in these registers can give you insight into your program's operation.

Moreover, GDB allows you to set breakpoints. This is like pausing a film to understand a crucial scene. Breakpoints let you pause program execution at specific places, helping you understand how your code works.

In summary, GDB is more than a debugging tool. It's a platform that enhances your ability to interact with and understand your software. Whether you're a developer, a cybersecurity enthusiast, or a coding

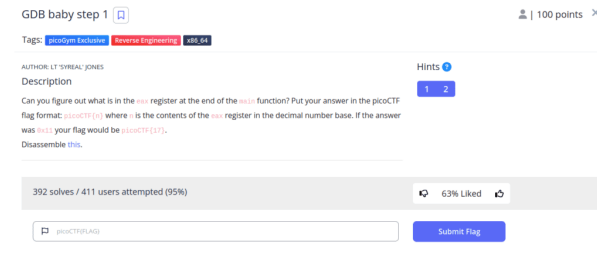
learner, GDB is a tool you'll want to learn.

So, if you're ready to explore programs in a new way, let's get started with GDB.

Preparations and File Analysis

Kickstart this adventurous journey by firing up your [Kali Linux virtual machine](#).

Navigate to the [PicoCTF](#) website, locate the challenge named "GDB Baby Step 1," in the "reverse engineering section", familiarize yourself with the task by reading the description, and download the file.



After downloading, shift this file into your dedicated workspace where we can take our time to analyse it. Curious to know more about the file?

So run:

```
$ file debugger0_a
debugger0_a: ELF 64-bit LSB pie executable, x86-64,
version 1 (SYSV), dynamically linked, interpreter /lib64/
ld-linux-x86-64.so.2,
BuildID[sha1]=15a10290db2cd2ec0c123cf80b88ed7d7f5cf9ff,
for GNU/Linux 3.2.0, not stripped
```

Voila! You have your answer. It's an Executable and Linkable Format (ELF) file, as you might have guessed. But thanks to that useful command you also know that it's a 64-bit executable and that's not stripped (so we can see its symbols).

The Magic of Disassembling with GDB

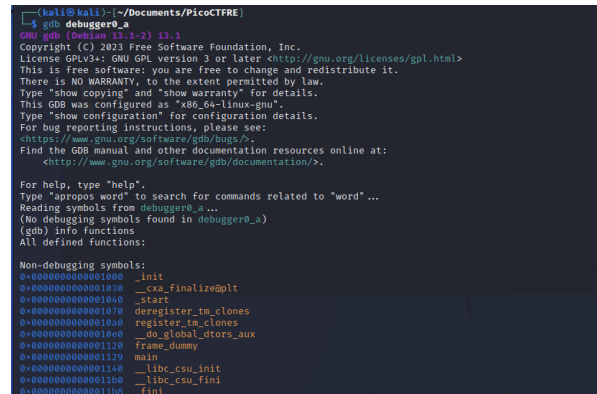
Enter the fascinating world of disassembly with GDB. Let's begin by opening the downloaded file by passing it as an argument to GDB:

```
gdb debugger0_a
```

Want to get an overview of all the functions? Just type:

```
info functions
```

As shown in the following screenshot, we have confirm that this file isn't stripped, and all function names are in plain sight.



Our prime target here is the 'main' function (address 0x1129).

Before we proceed, it's essential to know that GDB's syntax is set to AT&T

by default. Are you more comfortable with Intel syntax like most of us?

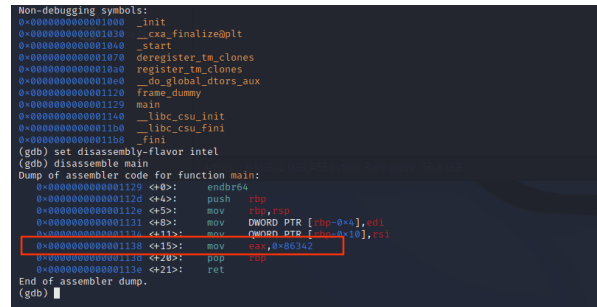
You can easily change the setting by typing:

```
set disassembly-flavor intel
```

Having done that, we're now set to uncover the assembly code by entering

```
disassemble main
```

The disassembly process unveils the following:



```
Non-debugging symbols:
0x0000000000001000 _init
0x0000000000001030 _cxa_finalize@plt
0x0000000000001040 _start
0x0000000000001070 deregister_tm_clones
0x0000000000001080 register_tm_clones
0x0000000000001090 do_global_ctors_aux
0x0000000000001120 frame_dummy
0x0000000000001120 main
0x0000000000001140 __libc_csu_init
0x0000000000001160 __libc_csu_fini
0x0000000000001180 _fini
(gdb) set disassembly-flavor intel
(gdb) disassemble main
Dump of assembler code for function main:
0x0000000000001129 <+0>: endbr64
0x000000000000112c <+4>: push rbp
0x000000000000112e <+5>: mov rbp,rbp
0x0000000000001131 <+8>: mov DWORD PTR [rbp-0x4],edi
0x0000000000001134 <+11>: mov QWORD PTR [rbp-0x10],rsi
0x0000000000001136 <+15>: mov ecx,0x86342
0x0000000000001138 <+17>: pop rbp
0x000000000000113e <+23>: ret
End of assembler dump.
(gdb)
```

The crux of this challenge is to discover the EAX register value. From the disassembled code, we see that the number 0x86342 is moved into this register.

However, the number is in hexadecimal format, and we need to convert it to a decimal format.

Let's embark on this conversion journey, that's an easy task with the help of our beloved python.

Open a new terminal and type

```
python
```

Then, to transform the hexadecimal value into a decimal, punch in:

```
print(int(0x86342))
```

Behold the decimal equivalent, 549698!

We've now arrived at our flag: **picoCTF{549698}**.

Just copy and paste this into the input field on the PicoCTF website.

Mission accomplished!

Alternative Path: Debugging the Executable

For those who love exploring alternative routes, let's solve this challenge through debugging.

Begin by setting a breakpoint at the main entry point.

Simply type:

```
break main
```

Next

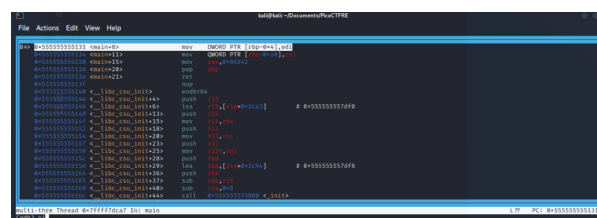
```
run
```

And then:

```
layout asm
```

for more detailed visualization.

Refer to the screenshot below for clarity:

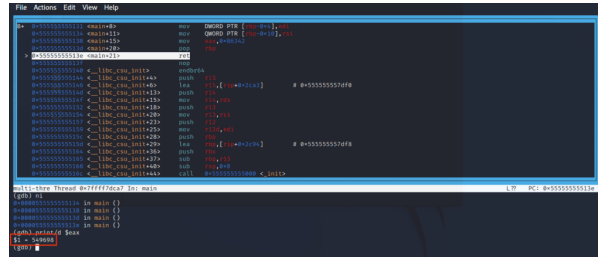


```
File Actions Edit View Help
[Debugger Window]
00401000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401001 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401002 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401003 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401004 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401005 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401006 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401007 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401008 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401009 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040100A 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040100B 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040100C 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040100D 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040100E 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040100F 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401010 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401011 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401012 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401013 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401014 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401015 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401016 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401017 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401018 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401019 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040101A 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040101B 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040101C 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040101D 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040101E 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040101F 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401020 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401021 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401022 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401023 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401024 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401025 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401026 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401027 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401028 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401029 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040102A 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040102B 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040102C 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040102D 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040102E 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040102F 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401030 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401031 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401032 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401033 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401034 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401035 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401036 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401037 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401038 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401039 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040103A 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040103B 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040103C 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040103D 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040103E 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040103F 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401040 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401041 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401042 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401043 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401044 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401045 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401046 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401047 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401048 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401049 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040104A 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040104B 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040104C 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040104D 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040104E 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040104F 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401050 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401051 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401052 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401053 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401054 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401055 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401056 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401057 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401058 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401059 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040105A 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040105B 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040105C 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040105D 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040105E 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040105F 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401060 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401061 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401062 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401063 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401064 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401065 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401066 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401067 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401068 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401069 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040106A 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040106B 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040106C 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040106D 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040106E 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040106F 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401070 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401071 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401072 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401073 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401074 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401075 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401076 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401077 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401078 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401079 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040107A 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040107B 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040107C 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040107D 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040107E 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040107F 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401080 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401081 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401082 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401083 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401084 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401085 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401086 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401087 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401088 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00401089 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040108A 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040108B 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040108C 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040108D 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040108E 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0040108F 00000000
```

for the next instruction, and then the “Enter” button until you reach the target instruction.

Now, reveal the EAX register’s decimal value by typing:

Here’s the screenshot displaying the value:

A screenshot of the GDB debugger interface. The top pane shows assembly code with instructions like 'movl \$0, %eax', 'leal 0x100000000, %eax', and 'addl \$0x1, %eax'. The bottom pane shows the GDB prompt '(gdb) ni' followed by '(gdb) print/d \$eax', which outputs '549698'. The status bar at the bottom indicates 'L7 RC: 0x00000000'.

Voila! As expected, the decimal value is 549698, corroborating our earlier findings.

This reaffirms that our flag remains steadfast at **picoCTF{549698}**.

Conclusion

To put it simply, GDB is an essential tool in your toolkit for reverse engineering. It’s crucial for taking apart and investigating code, making it an invaluable resource for solving puzzles like PicoCTF’s “GDB Baby Step 1”. This journey we’ve taken, uncovering the value in the EAX register, making sense of the ELF file, and cracking the challenge is a perfect example of GDB’s power.

Are you feeling a rush of achievement? That’s the beauty of learning! But remember, this is just the beginning. The field of reverse engineering is vast and full of endless mysteries waiting to be unlocked.

Are you eager to keep exploring? We’re excited to share that journey with you. Stay in the loop with all the latest learning opportunities and challenges by following us at [Stackzero](#) and connecting with us on our social media profiles.

Remember, the world of learning is vast and open to all. Every expert was once a beginner too. So, keep exploring, keep learning, and continue to uncover the exciting puzzles that reverse engineering offers. I think that CTFs are the best way to learn and have fun at the same time.

The more you learn, the more fascinating it becomes. So, let’s keep that curiosity alive and dive deeper into the world of reverse engineering together!