



Binary Search

Binary Search



Easy **General Skills** **picoCTF 2024** **shell** **browser_webshell_solvable** **ls**

AUTHOR: JEFFERY JOHN

Description

Want to play a game? As you use more of the shell, you might be interested in how they work! Binary search is a classic algorithm used to quickly find an item in a sorted list. Can you find the flag? You'll have 1000 possibilities and only 10 guesses.

Cyber security often has a huge amount of data to look through - from logs, vulnerability reports, and forensics. Practicing the fundamentals manually might help you in the future when you have to write your own tools!

You can download the challenge files here:

- [challenge.zip](#)

Additional details will be available after launching your challenge instance.

This challenge launches an instance on demand.

Its current status is:

NOT_RUNNING

[Launch Instance](#)

Hints

1 **2** **3**

The program will randomly choose a new number each time you connect. You can always try again, but you should start your binary search over from the beginning - try around 500. Can you think of why?

Author: The Analyst: Hyposelenia

Challenge: Binary Search

Category: General Skills

Date: 01/15/26



I. Objective

The objective of this challenge was to obtain the hidden flag by correctly guessing a number generated by a remote server using the **binary search technique** within a limited number of attempts.

II. Background

Binary search is a classic algorithm used to efficiently find a value within a **sorted range** by repeatedly dividing the search space in half. Instead of guessing numbers randomly, binary search uses feedback such as “**higher**” or “**lower**” to eliminate half of the remaining possibilities after each guess.

In this challenge, the server generated a number between **1 and 1000** and allowed only **10 guesses**, which strongly implies that binary search is the intended solution.

III. Tool Used

- **Oracle VirtualBox (Kali Linux)** – Used as the Linux environment
- **Linux Terminal** – Used to execute decoding commands
- **ssh (Secure Shell)** – Used to securely connect to the challenge instance

IV. Methodology

1. Opened Oracle VirtualBox and launched Kali Linux (the picoCTF webshell may also be used).
2. Connected to the remote challenge instance using SSH:

```
ssh -p <port> <username>@<hostname>
```

3. Accepted the SSH fingerprint by typing yes.
4. Entered the provided password (note: passwords do not display while typing for security reasons).



5. Used the **binary search strategy** by guessing the midpoint of the current range.
6. Adjusted guesses based on the server's feedback ("Higher" or "Lower").
7. Repeated the process until the correct number was found.
8. The flag was displayed after successfully guessing the number.

V. Result

picoCTF{g00d_gu355_de9570b0}

```
(kali㉿kali)-[~]
└─$ ssh -p 64420 ctf-player@atlas.picocft.net
The authenticity of host '[atlas.picocft.net]:64420 ([18.217.83.136]:64420)' can't be established.
ED25519 key fingerprint is SHA256:M8hXanE8l/Yzfs8iuxNsuFL4vCzCKEIlM/3hp013tfQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[atlas.picocft.net]:64420' (ED25519) to the list of known hosts.
ctf-player@atlas.picocft.net's password:
Welcome to the Binary Search Game!
I'm thinking of a number between 1 and 1000.
Enter your guess: 500
Lower! Try again.
Enter your guess: 250
Higher! Try again.
Enter your guess: 300
Lower! Try again.
Enter your guess: 280
Higher! Try again.
Enter your guess: 290
Lower! Try again.
Enter your guess: 287
Lower! Try again.
Enter your guess: 283
Higher! Try again.
Enter your guess: 285
Higher! Try again.
Enter your guess: 286
Congratulations! You guessed the correct number: 286
Here's your flag: picoCTF{g00d_gu355_de9570b0}
Connection to atlas.picocft.net closed.

(kali㉿kali)-[~]
└─$
```

VI. Explanation



This challenge demonstrates how **binary search works in practice**. At the start, there are **1000 possible numbers**. Each guess divides the possible range into **two halves**:

- If the server replies “**Lower**”, all higher numbers are discarded.
- If the server replies “**Higher**”, all lower numbers are discarded.

By always guessing the **middle number** of the remaining range, the search space is reduced by half every time. Because binary search reduces possibilities exponentially, it guarantees finding the correct number within **10 guesses**, which matches the challenge limit.

This method is far more efficient than random guessing and is commonly used in computing and cybersecurity when searching large datasets quickly.

VII. Conclusion

The Binary Search challenge reinforces the importance of algorithmic thinking in cybersecurity. By applying the binary search technique, the correct number was found efficiently within the allowed attempts, resulting in the successful retrieval of the flag.

Understanding binary search is valuable not only for programming but also for real-world cybersecurity tasks such as log analysis, vulnerability testing, and data searching.

— The Analyst: Hyposelenia