Final Project Report

Xin He, U81612345
Yang Yang, U09302475
Haoning Wu, U19366249
Maoxuan Zhu U24023009

Github link: https://github.com/mmao95/CS655-GENIMINI
GENI slice name: Cracker
GENI slice link:
https://portal.geni.net/secure/slice.php?slice_id=19cb9751-aeab-4d68-9d1e-920227cb8748

Password Cracker (Serverless)

Introduction:

Every boy has a dream to become a hacker so we decided to implement the Password Cracker.

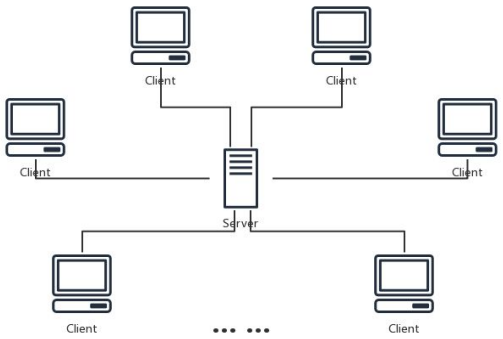
This is a distributed system where a user submits the md5 hash of a 5-character password (a-z, A-Z) to the system using a web interface. The web interface with the help of worker nodes cracks the password by a brute force approach.

Our password cracker system has a server which is responsible for dividing the tasks to different parts and client which is able to brute force a specific task.

We also used the serverless methodology to simplify the client-side work and make our program easier to use.

Experimental Methodology:

- Assumption:
    1. Our server will never crash.
    2. Clients can crash and it is running initially and trying to reach the server.
    3. Clients normally will not crash so we gave each client 4 segments of password to guess.
    4. We only have one password to crack at one time.
    5. We will randomly generate a password on the server-side.
- Architecture Diagram:

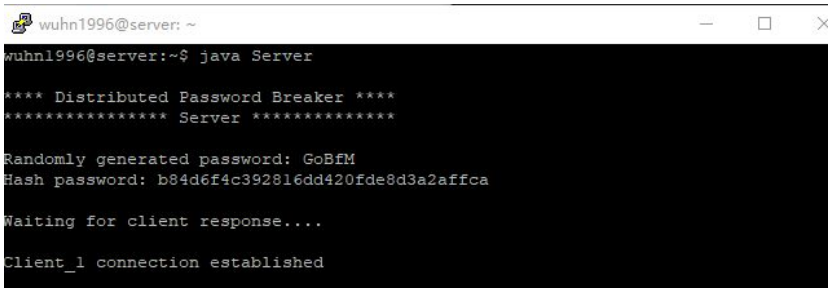


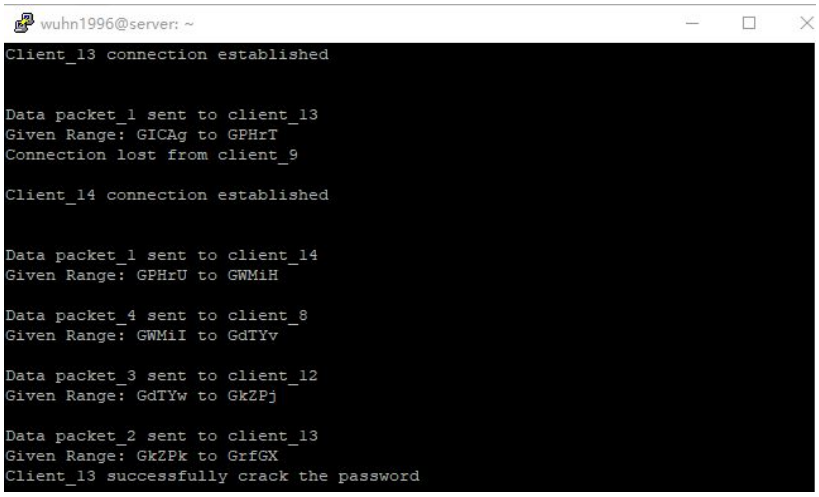
Result:

- Usage instruction:
    1. Install JDK on GENI Node
    2. Run server code first to simulate the running server
    3. Since it is a scalable distributed password cracker, we can run client at any time.
    4. Our system is fault-tolerance, so if any client crushed other idle nodes will continue crashing the password.
- Analysis:

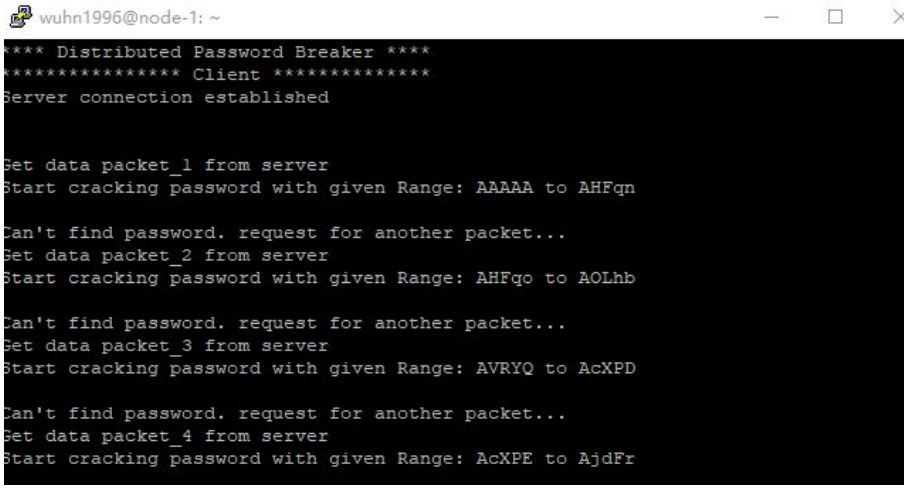
Followings are the screenshots from the Server and Clients during a password cracking process:

Server:

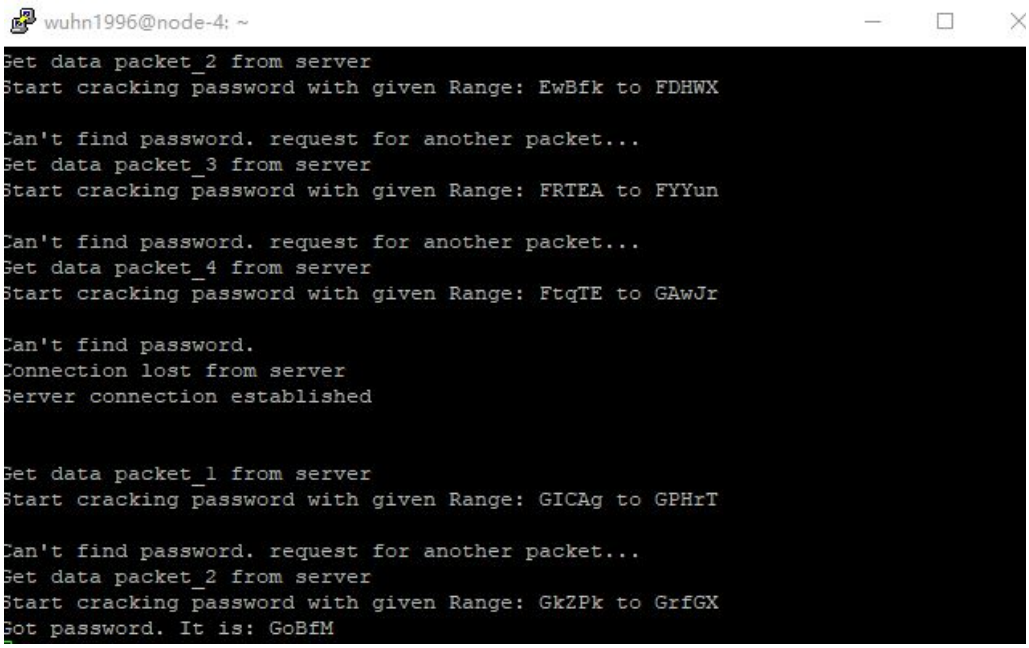



wuhn1996@server: ~
Client_13 connection established
Data packet_1 sent to client_13
Given Range: GICAg to GPHrT
Connection lost from client_9
Client_14 connection established
Data packet_1 sent to client_14
Given Range: GPHrU to GWMiH
Data packet_4 sent to client_8
Given Range: GWMiI to GdTYv
Data packet_3 sent to client_12
Given Range: GdTYw to GkZPj
Data packet_2 sent to client_13
Given Range: GkZPk to GrfGX
Client_13 successfully crack the password

Clients:


wuhn1996@node-1: ~
**** Distributed Password Breaker ****
********** Client **********
Server connection established
Get data packet_1 from server
Start cracking password with given Range: AAAAA to AHFqn
Can't find password. request for another packet...
Get data packet_2 from server
Start cracking password with given Range: AHFqo to AOLhb
Can't find password. request for another packet...
Get data packet_3 from server
Start cracking password with given Range: AVRYQ to AcXPD
Can't find password. request for another packet...
Get data packet_4 from server
Start cracking password with given Range: AcXPE to AjdFr


wuhn1996@node-2: ~
Server connection established
Get data packet_1 from server
Start cracking password with given Range: DkBKo to DrHBb
Can't find password. request for another packet...
Get data packet_2 from server
Start cracking password with given Range: EhpyI to Eovov
Can't find password. request for another packet...
Get data packet_3 from server
Start cracking password with given Range: FYYuo to Ffelb
Can't find password. request for another packet...
Get data packet_4 from server
Start cracking password with given Range: GWMiI to GdTYv
Can't find password.
Connection lost from server



According to the screenshot, the password generated is GoBfM. Once a Client sets up a connection with the Server, it will be assigned amount of possible passwords. If any of them is the password generated by the user, the Client will report it and disconnect with the Server immediately, otherwise, it will continue attempting to connect to the Server and waiting for new tasks. In this experiment, Client 4 finds the correct password and reports it to the Server.

Conclusion:

We come to understand why all the passwords on the internet only allow users to try five times. The reason is that the brute-force way of password cracker is actually very efficient with the help of the network.

Division of Labors:

Maoxuan Zhu: Server code, the experimental methodology part of the report
Haoning Wu: Server code, the result part of the report
Xin He: Client code, the conclusion part of the report
Yang yang: Client code, the Introduction part of the report