- *Project Title:*
- Personal firewall using python:
- *Objective:*
- To simulate a personal firewall that inspects incoming IP address and determines whether to allow or block the connection based on predefined rules.
- *Tools & Technologies Used:*
- Programming language: python3
- Module Used: random
- *Project Description:*
- This python project emulates the working of a personal firewall. It generates a random IP address from a specific range and then checks this IP address against a set of firewall rules.
- If the generated IP address matches any of the blocked IPs define in the rules set the connection is blocked; otherwise, it is allowed. additionally, a random number is generated for each request to simulate tracking or logging purposes.
- Key Functionalities:

1. *Random IP Generation:*

```
Python
Def  generate_random_ip();
        return  f"192.168.1.{random.randint(0, 20)}"
```

- Produce IPs in the range 192.168.1.0 to 192.168.1.20.

2. *Firewall Rule Checker:*

```
Python
Def check_firewall_rules(ip,  rules) :
     for rule_ip,  sction in rules.items():
          if ip == rule_ip:
              return action
return "allow"
```

- Compares the IP to the dictionary of blocked IPs.

3. *Simulation Logic:*

```
Python

for  _  in range(12):
```

- Simulates 12 attempts, displaying the action and a random tracking number.

- *Firewall Rules in Code:*

Python

```
Firewall_rules= {
    “192.168.1.1.”: “block”,
    “192.168.1.4.”: “block”,
    “192.168.1.9.”: “block”,
    “192.168.1.13”: “block”,
    “192.168.1.16”: “block”,
    “192.168.1.19.”: “block”,
}
```

These are IP address that the firewall will block if encountered.

Sample Output:

yaml

```
Ip: 192.168.1.16, Action: block, Random: 7392
IP: 192.168.1.15, Action: block, Random: 2540
Ip: 192.168.1.9, Action: block, Random: 8285
```

Each line represents a connection attempt:

- Ip address generated
- Firewall's decision (allow or block)
- Random number (simulation ID)

- *Conclusion:*

This project is a simple yet powerful way to understand how firewall rules operate. It shows how specific IPs can be blocked while others are allowed, marking it a great beginner-level project in network security simulation using python.

- ***Future Enhancements:***

- Allow input of firewall rules via a config file or user input.
- Support for port numbers and protocols (TCP/UDP).
- Logging results to a file.
- Adding a GUI interface using PyQt.
- Detect and block IPs using CIDR or subnetting.