|  |  |
| --- | --- |
| I2NSF Hackathon Project | Translator Project |
| Simulated Social networking Server (Facebook) | Considers the IP addresses of actual DNS servers of websites (Facebook, Gmail etc )  Advantage: Helps to block the DNS servers and hence terminating the traffic inflow from them in real time.  Disadvantage: The IP address of the website’s DNS server has to be cached in our server environment.  (Simulation to ping the website’s DNS server and find it’s ip address is being developed) |
| Development language: Python | Java |
| Client facing interface: Web UI as interface for giving policy inputs | Currently using Eclipse IDE console to input policies  WebUI has been built to input high level policies and to translate them to Security rules as part of thesis published, “Network Security tool for a Novice”. |
| Uses Mininet to simulate SDN environment | Uses VM environment deployed in Huawei’s Server  Uses OpenStack environment to provision VMs for NSFs |
| Considers Time dependent firewall | Firewall considered – pfSense, Netfilter  IDS considered - Snort |
| Works for Update and deletion of policy as well | Currently works with insertion of policy  Update and deletion of policy is being developed |
| SDN topology was the main focus | Traditional network environment is mainly considered  Working on extending to SDN environment |
| Implemented use case to work with VoIP/VoLTE | Not considered |
| Feasibility of Data Driven Approach: YANG | Working on compatibility of YANG data models working with NSFs |

**Comparison of I2NSF Hackathon project and Our implementation**

**Translator Project – High level architecture**

1. Input YANG file: “file.yang”

2. Translate Input file

Security Controller

4. Implement the YANG file in NSF (Firewall, IDS)

NSF

3. Translated Input YANG file: “file.yang”