I2NSF Framework Project @ IETF-101 Hackathon

IETF 101, London

March 18, 2018

Champions: Jaehoon Paul Jeong and Jinyong Tim

Sungkyunkwan University

Why Did We Do this Project?

- I2NSF: Use NETCONF, RESTCONF, YANG Data Models
  - Is I2NSF reasonable for the management of network security functions?
  - Can we implement I2NSF using open source software?

- This work is a student project!!
  - 7 graduate students at Sungkyunkwan University
  - Source Code on Github
    - https://github.com/kimjinyong/i2nsf-framework/tree/master/Hackathon-101

IETF I2NSF (Interface to Network Security Functions) Working Group: I2NSF Framework Project

Champions: Jaehoon Paul Jeong and Jinyong Tim Kim (SKKU)

Professors

- Jaehoon (Paul) Jeong (Sungkyunkwan)
- Hyoungshick Kim (Sungkyunkwan)
- Sangwon Hyun (Sungkyunkwan)

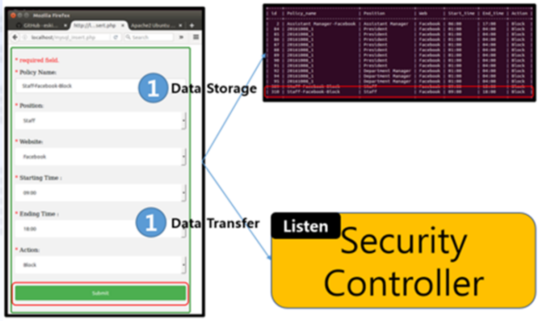
Collaborators

- Jung-Soo Park (ETRI)
- Tae-Jin Ahn (Korea Telecom)

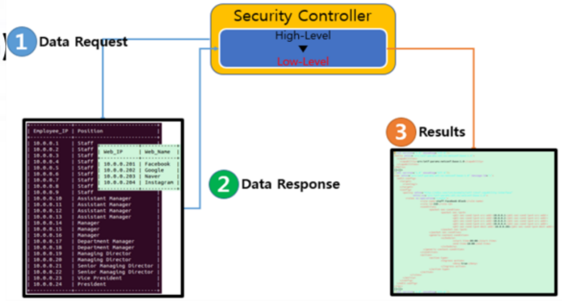
Students

- Jinyong Tim Kim
- Eunsoo Kim
- Dongjin Hong
- Tae-Kyun Roh
- Sarang Wi
- Seungjin Lee
- Jinhyuk Yang

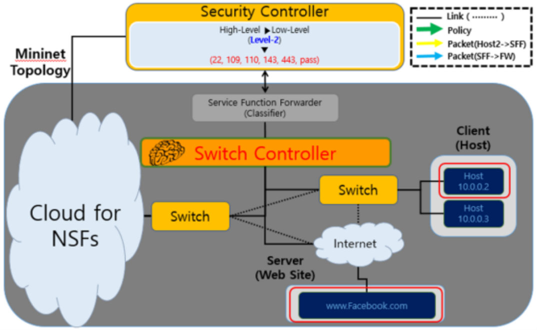
I2NSF User (Web)



Security Controller



Network Security Functions (NSF) – Triggered Steering



Where to get code

- Github – Source code
  - ✓ https://github.com/kimjinyong/i2nsf-framework

What to pull down to set-up environment

- OS: Ubuntu 14.04TL
- Confd for NETCONF: 6.2 Version
- Apache2: 2.4.7 Version
- MySQL: 14.14 Version
- PHP: 5.5.9 Version
- Mininet: 2.2.1 Version
- OpenDaylight: Distribution-karaf-0.4.3-Beryllium-SR3
- XSLT (Extensible StyleSheet Languages Transformations)
- Jetconf: Jetconfis a python Open APIfor RESTCONF.

Manual for Operation Process

- README.txt

Contents of Implementation

- I2NSF Framework for provisioning Network Security Functions (NSF)
  - ✓ Consumer-Facing Interface via RESTCONF/YANG (New Feature)
  - ✓ NSF-Facing Interface via NETCONF/YANG
  - ✓ Registration Interface via NETCONF/YANG
- Network Security Functions
  - ✓ Firewall using SDN and Suricata
  - ✓ Deep Packet Inspection (DPI) using Suricata
- Advanced Functions
  - ✓ Dynamic Policy Configuration (New Feature)
  - ✓ NSF-triggered Traffic Steering using SFC (New Feature)
  - ✓ YANG Data Modeling for NSF Monitoring

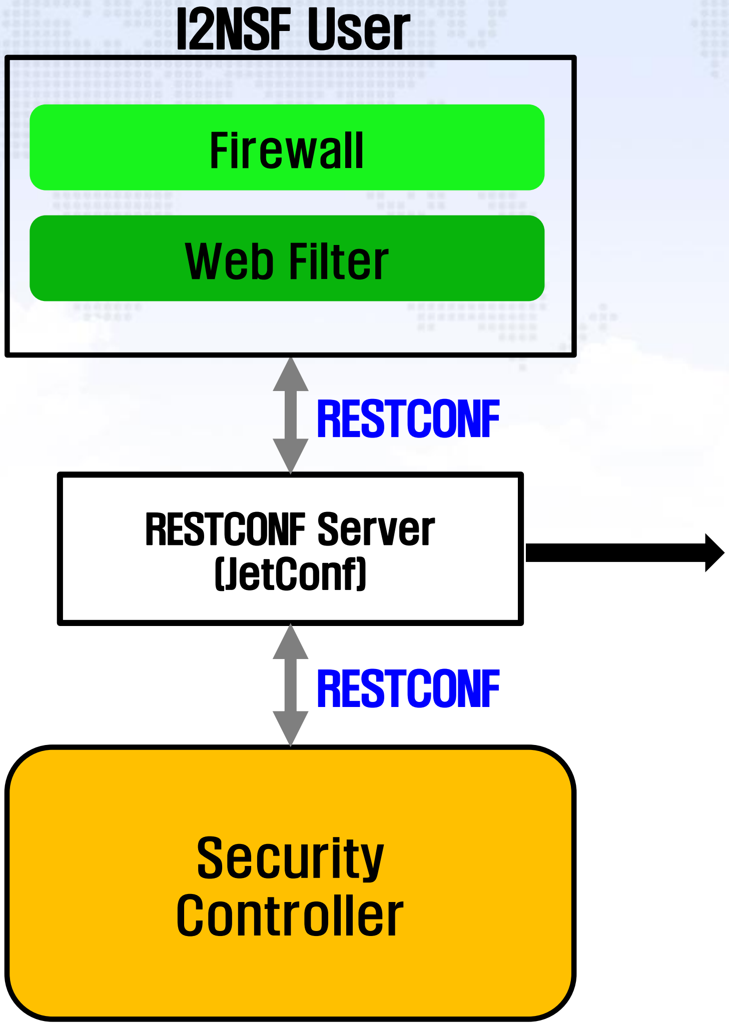
Goal of I2NSF Project

I2NSF Framework is extended with

1. Dynamic Configuration to map Security Service to Network Security Function at Security Controller.

2. Consumer-Facing Interface based on RESTCONF and the latest YANG Data Model.

I2NSF Consumer-Facing Interface



I2NSF User

Firewall

Web Filter

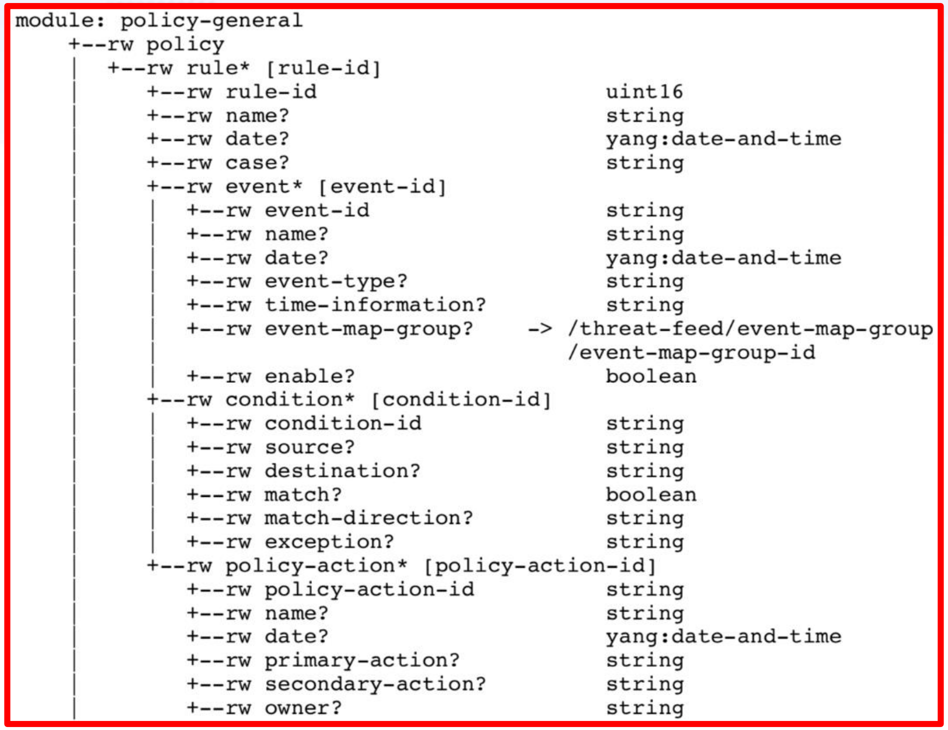
RESTCONF

RESTCONF Server (JetConf)

RESTCONF

Security Controller

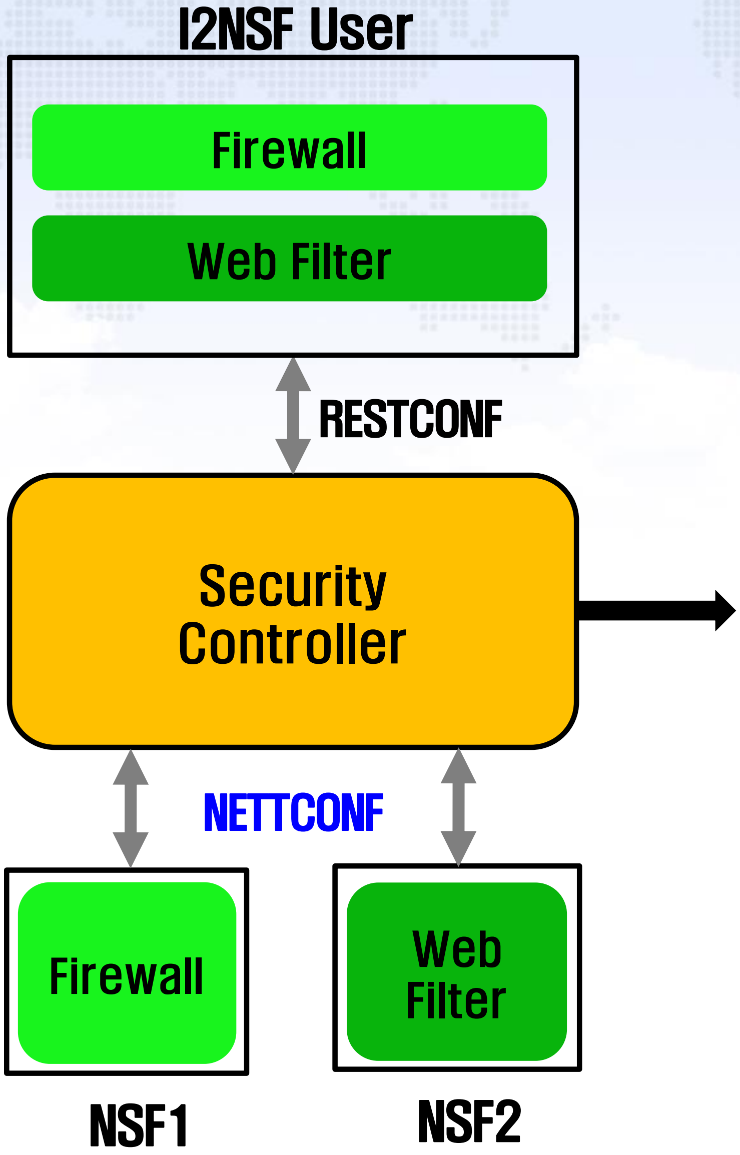
YANG Data Model for Consumer-Facing Interface: draft-ietf-i2nsf-consumer-facing-interface-dm-00



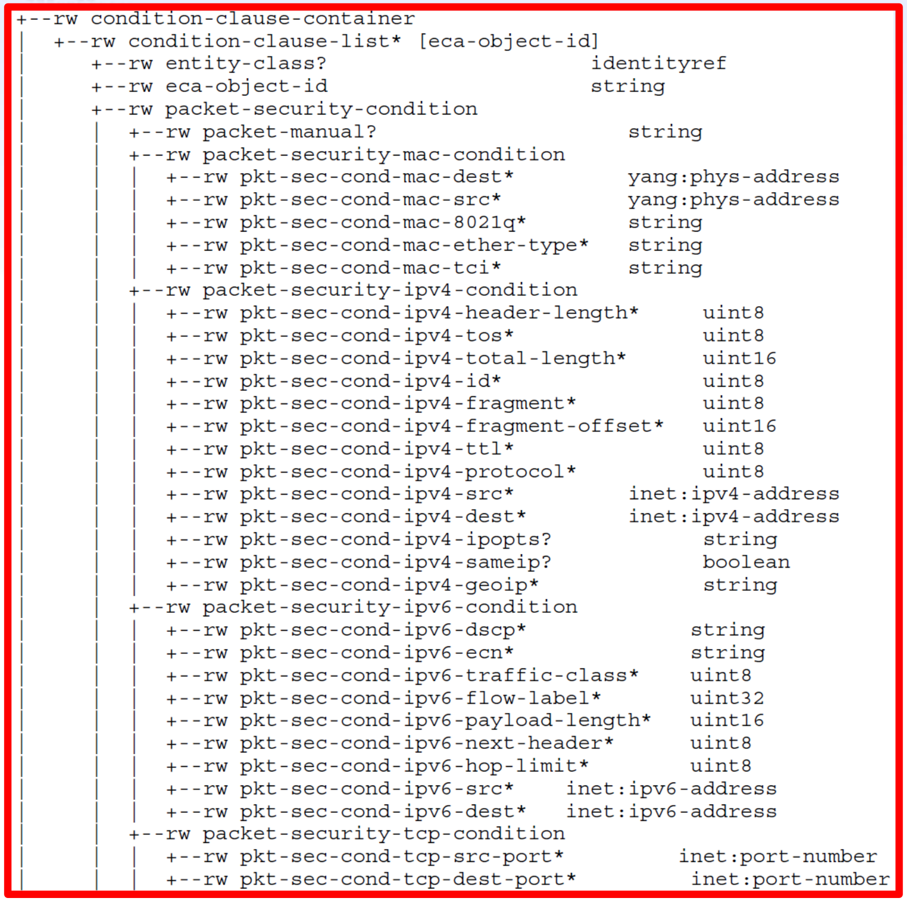
```
module: policy-general
    +--rw policy
    |  +--rw rule* [rule-id]
    |     +--rw rule-id                      uint16
    |     +--rw name?                        string
    |     +--rw date?                        yang:date-and-time
    |     +--rw case?                        string
    |     +--rw event* [event-id]
    |     |  +--rw event-id                   string
    |     |  +--rw name?                      string
    |     |  +--rw date?                      yang:date-and-time
    |     |  +--rw event-type?                string
    |     |  +--rw time-information?          string
    |     |  +--rw event-map-group?    -> /threat-feed/event-map-group
    |     |                               /event-map-group-id
    |     |  +--rw enable?                    boolean
    |     +--rw condition* [condition-id]
    |     |  +--rw condition-id               string
    |     |  +--rw source?                    string
    |     |  +--rw destination?               string
    |     |  +--rw match?                     boolean
    |     |  +--rw match-direction?           string
    |     |  +--rw exception?                 string
    |     +--rw policy-action* [policy-action-id]
    |        +--rw policy-action-id           string
    |        +--rw name?                      string
    |        +--rw date?                      yang:date-and-time
    |        +--rw primary-action?            string
    |        +--rw secondary-action?          string
    |        +--rw owner?                     string
```

Enhanced Security Policy Delivery having Event-Condition-Action Paradigm to Security Controller via RESTCONF Server

I2NSF NSF-Facing Interface



YANG Data Model for NSF-Facing Interface: draft-ietf-i2nsf-nsf-facing-interface-dm-00



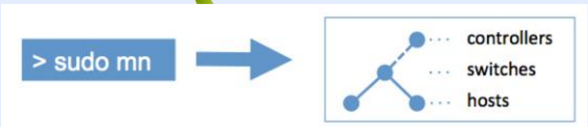
```
+--rw condition-clause-container
|  +--rw condition-clause-list* [eca-object-id]
|     +--rw entity-class?                        identityref
|     +--rw eca-object-id                        string
|     +--rw packet-security-condition
|     |  +--rw packet-manual?                    string
|     |  +--rw packet-security-mac-condition
|     |  |  +--rw pkt-sec-cond-mac-dest*         yang:phys-address
|     |  |  +--rw pkt-sec-cond-mac-src*          yang:phys-address
|     |  |  +--rw pkt-sec-cond-mac-8021q*        string
|     |  |  +--rw pkt-sec-cond-mac-ether-type*   string
|     |  |  +--rw pkt-sec-cond-mac-tci*          string
|     |  +--rw packet-security-ipv4-condition
|     |  |  +--rw pkt-sec-cond-ipv4-header-length*     uint8
|     |  |  +--rw pkt-sec-cond-ipv4-tos*               uint8
|     |  |  +--rw pkt-sec-cond-ipv4-total-length*      uint16
|     |  |  +--rw pkt-sec-cond-ipv4-id*                uint8
|     |  |  +--rw pkt-sec-cond-ipv4-fragment*          uint8
|     |  |  +--rw pkt-sec-cond-ipv4-fragment-offset*   uint16
|     |  |  +--rw pkt-sec-cond-ipv4-ttl*               uint8
|     |  |  +--rw pkt-sec-cond-ipv4-protocol*          uint8
|     |  |  +--rw pkt-sec-cond-ipv4-src*         inet:ipv4-address
|     |  |  +--rw pkt-sec-cond-ipv4-dest*        inet:ipv4-address
|     |  |  +--rw pkt-sec-cond-ipv4-ipopts?            string
|     |  |  +--rw pkt-sec-cond-ipv4-sameip?            boolean
|     |  |  +--rw pkt-sec-cond-ipv4-geoip*             string
|     |  +--rw packet-security-ipv6-condition
|     |  |  +--rw pkt-sec-cond-ipv6-dscp*             string
|     |  |  +--rw pkt-sec-cond-ipv6-ecn*              string
|     |  |  +--rw pkt-sec-cond-ipv6-traffic-class*    uint8
|     |  |  +--rw pkt-sec-cond-ipv6-flow-label*       uint32
|     |  |  +--rw pkt-sec-cond-ipv6-payload-length*   uint16
|     |  |  +--rw pkt-sec-cond-ipv6-next-header*      uint8
|     |  |  +--rw pkt-sec-cond-ipv6-hop-limit*        uint8
|     |  |  +--rw pkt-sec-cond-ipv6-src*    inet:ipv6-address
|     |  |  +--rw pkt-sec-cond-ipv6-dest*   inet:ipv6-address
|     |  +--rw packet-security-tcp-condition
|     |  |  +--rw pkt-sec-cond-tcp-src-port*         inet:port-number
|     |  |  +--rw pkt-sec-cond-tcp-dest-port*         inet:port-number
```

Dynamic Configuration for the Mapping from Security Service to NSF

Hackathon Development

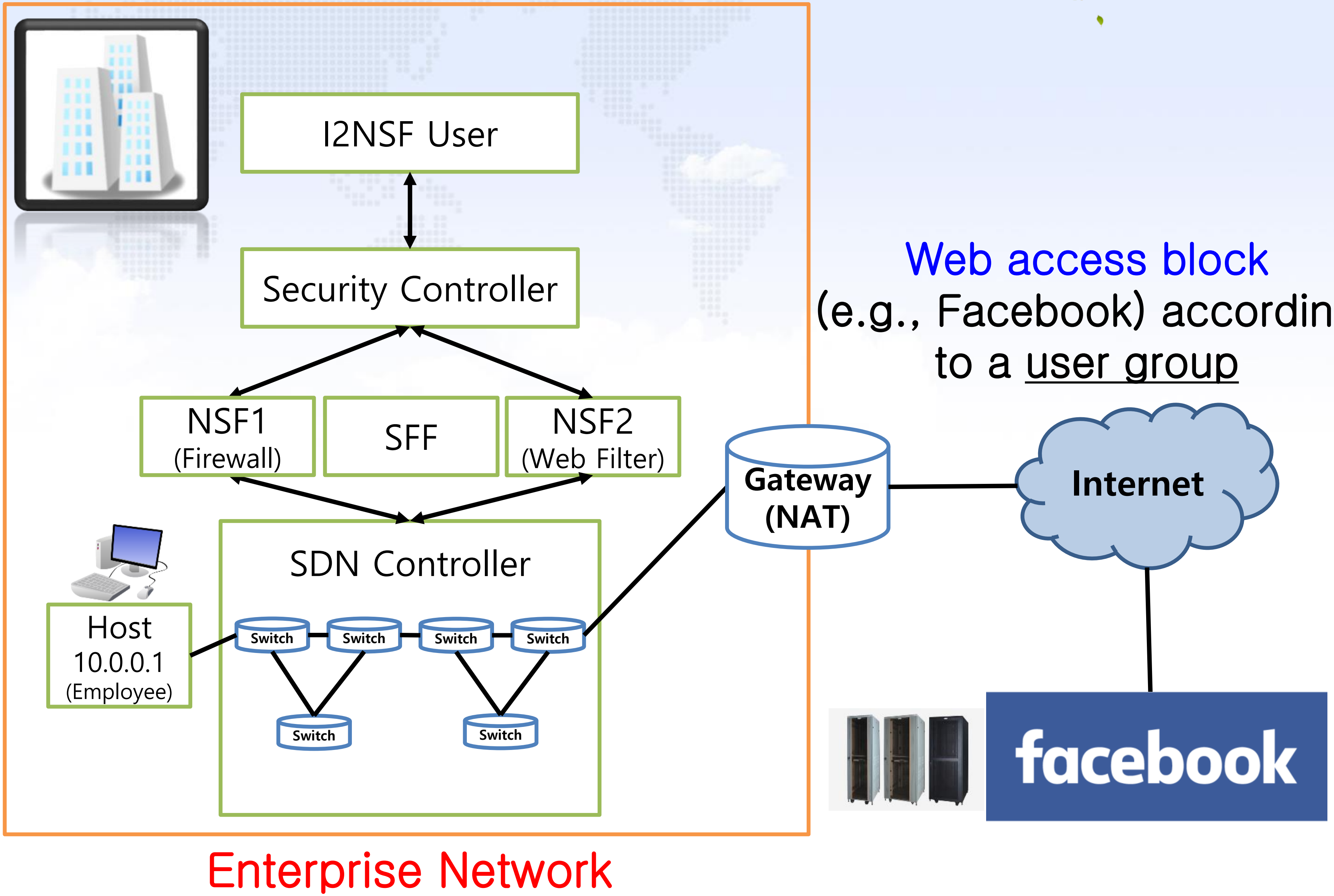
Build Environment

1. OS
   - Ubuntu 14.04TL
2. Netconfd
   - 6.2 Version
3. Apache2
   - 2.4.7 Version
4. MySQL
   - 14.14 Version
5. PHP
   - 5.5.9 Version



5. Mininet
   - 2.2.1 Version
6. OpenDaylight
   - Distribution-karaf-0.4.3-Beryllium-SR3
7. Suricata
   - 3.2.1 RELEASE

Network Configuration for Hackathon

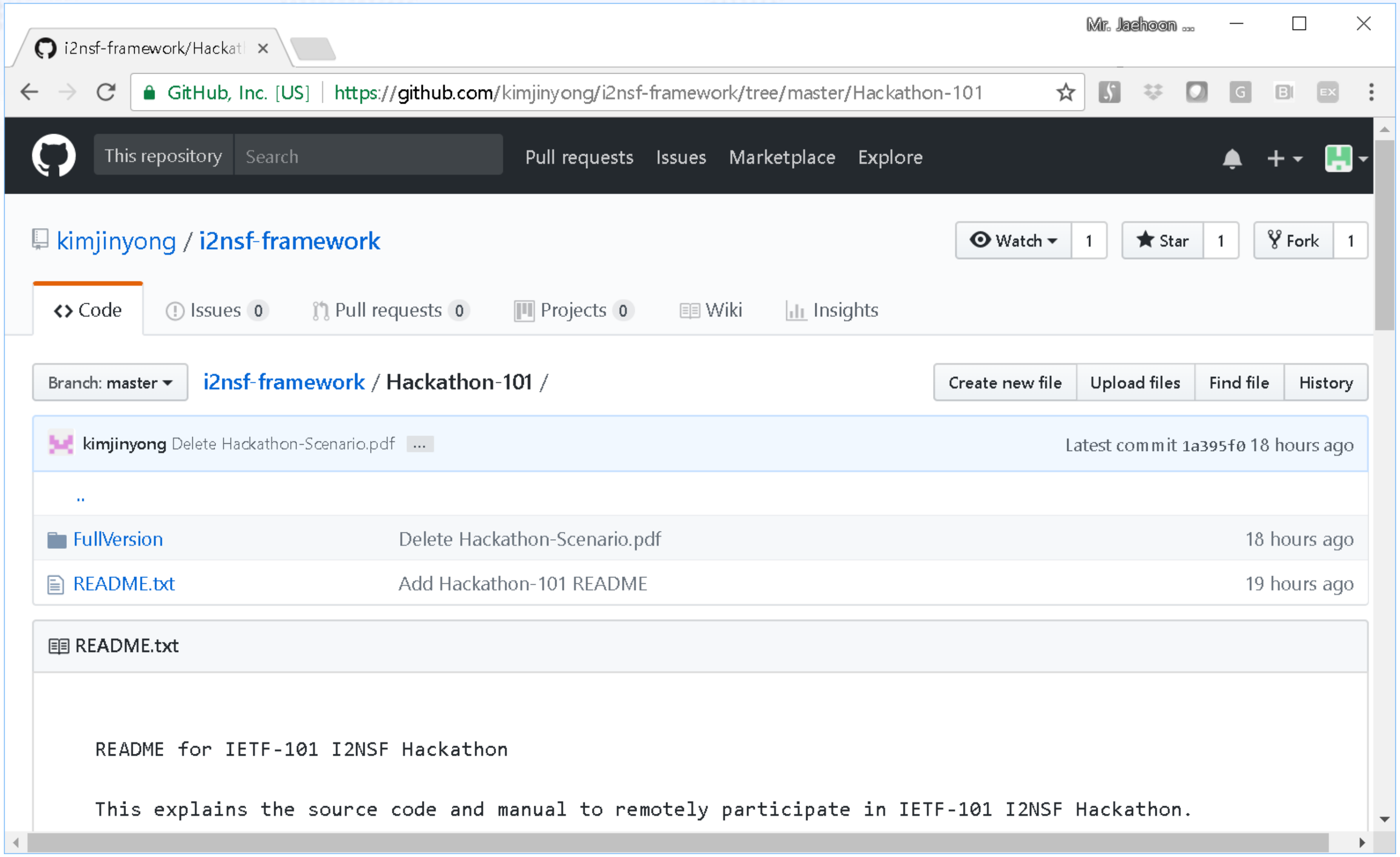


Web access block (e.g., Facebook) according to a user group

Information of I2NSF Hackathon Project

Github for I2NSF Framework Project

- **Documents and Source Code**
  https://github.com/kimjinyong/i2nsf-framework/tree/master/Hackathon-101



i2nsf-framework / Hackathon-101 /

kimjinyong Delete Hackathon-Scenario.pdf ... Latest commit 1a395f0 18 hours ago

..

FullVersion	Delete Hackathon-Scenario.pdf	18 hours ago
README.txt	Add Hackathon-101 README	19 hours ago

README.txt

```
README for IETF-101 I2NSF Hackathon

This explains the source code and manual to remotely participate in IETF-101 I2NSF Hackathon.
```

Lessons from the Implementation @ Hackathon

- **Proof of Concept (POC) of I2NSF Framework and YANG Data Models using Open Sources:**
  - **Confd for I2NSF NSF-Facing and Registration Interface**
  - **JetConf for Consumer-Facing Interface**
  - **Suricata for NSFs (i.e., Firewall and Web Filter)**
  - **OpenDaylight for SDN Controller**
  - **NSH and tunneling for packet steering over NSFs**
  - **Mininet for SDN Network**