IETF-114

I2NSF Hackathon Project

July 23-24, 2022

Champion: Jaehoon (Paul) Jeong

Members: Patrick Lingga and Jeonghyeon Kim

Department of Computer Science and Engineering at SKKU

I2NSF (Interface to Network Security Functions) Framework Project

Champion: Jaehoon (Paul) Jeong

I2NSF Hackathon Project

Professors:

- Jaehoon (Paul) Jeong (SKKU)
- Younghan Kim (SSU)

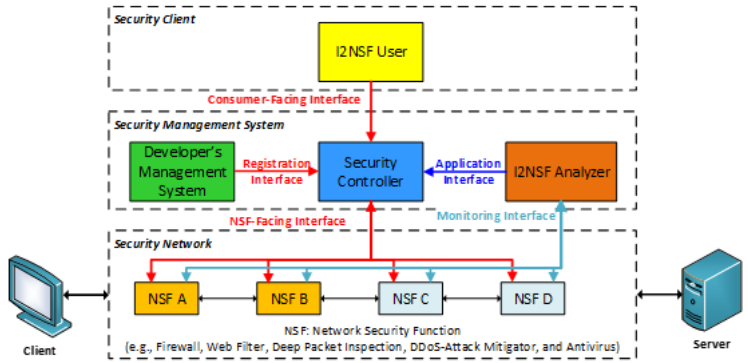
Researchers:

- Jung-Soo Park (ETRI)
- Yunchul Choi (ETRI)
- Jinyong Kim (SKKU)

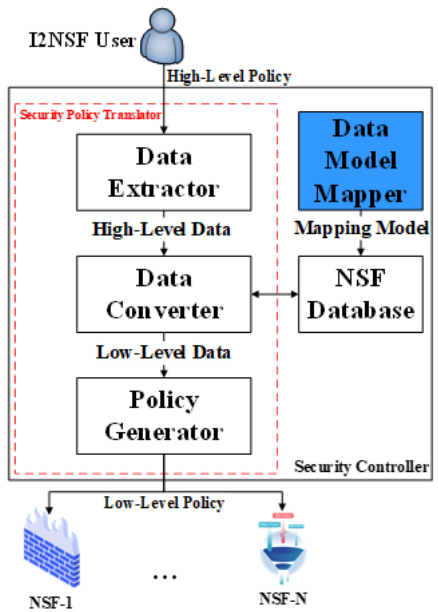
Students:

- Patrick Lingga (SKKU)
- Jeonghyeon Kim (SKKU)
- Hadong Park (Calvin University)

I2NSF Framework



I2NSF Security Policy Translator



Where to get Code and Demo Video Clip

- Github – Source Code
  - ✓ https://github.com/jaehoonpaul/i2nsf-framework

What to pull down to set up an environment

- OS: Ubuntu 16.04 LTS
- ConfD for NETCONF: 6.6 Version
- Jetconf for RESTCONF
- OpenStack: Queens version
- NSF: Suricata
- Hyperledger Fabric: 2.2 version

Manual for Operation Process

- I2NSF-Manual-Hackathon.md contains detailed description about operation process. It can be found in the GitHub.

Contents of Implementation

- Cloud-based Security Service System using I2NSF Framework
  - ✓ Web-based I2NSF User
  - ✓ Console-based Security Controller
  - ✓ Console-based Developer's Management System
  - ✓ I2NSF Framework in OpenStack NFV Environment
  - ✓ I2NSF Capability YANG Data Model
  - ✓ Registration Interface via NETCONF/YANG
  - ✓ Consumer-Facing Interface via RESTCONF/YANG
  - ✓ NSF-Facing Interface via NETCONF/YANG
  - ✓ Monitoring Interface via NETCONF/YANG
  - ✓ Web-based NSF Monitoring
  - ✓ Application Interface as Feedback from I2NSF Analyzer
- Network Security Functions
  - ✓ Firewall and Web-filter using Suricata
- Advanced Functions
  - ✓ Security Policy Translation: Automatic Generation of Low-Level Policy with Policy Provisioning
  - ✓ Blockchain-based Auditing for I2NSF Policy and Data Transactions

성균관대학교 SUNG KYUN KWAN UNIVERSITY ETRI openstack. HYPERLEDGER FABRIC ConfD tail-f a Cisco company SURICATA

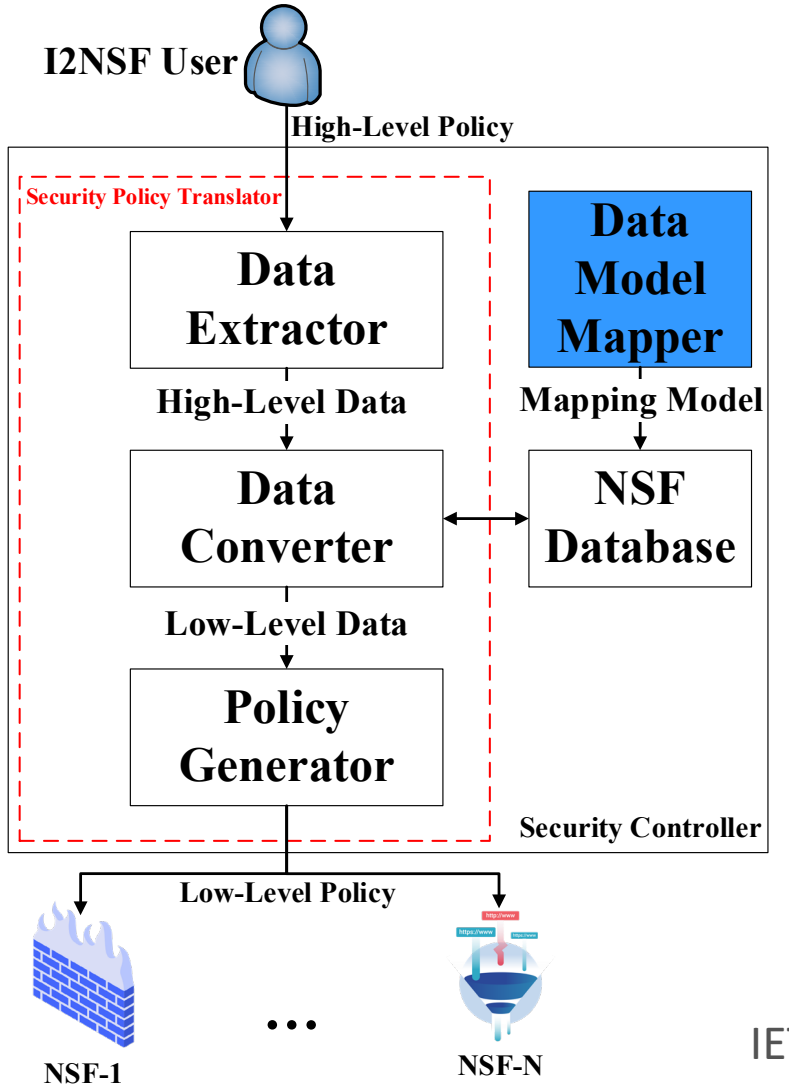
Hackathon Plan (1/2)

- The Implementation of the Internet Drafts for the I2NSF System for Cloud-based Security Services:
  - draft-ietf-i2nsf-capability-data-model-32
  - draft-ietf-i2nsf-consumer-facing-interface-dm-21
  - draft-ietf-i2nsf-nsf-facing-interface-dm-29
  - draft-ietf-i2nsf-registration-interface-dm-18
  - draft-ietf-i2nsf-nsf-monitoring-data-model-20
  - draft-yang-i2nsf-security-policy-translation-11
  - draft-jeong-i2nsf-security-management-automation-03

- Implementation of Security Policy Translator in I2NSF Framework.

Hackathon Plan (2/2)

- Implementation of Security Policy Translator



The overall architecture of our scheme consists of five components:

- ✓ Data Extractor.
- ✓ Data Converter.
- ✓ NSF Database.
- ✓ Policy Generator.
- ✓ Data Model Mapper.

What got done (1/4)

- Data Model Mapper Results:

Consumer-Facing Interface's YANG Data Model Attributes

NSF-Facing Interface's YANG Data Model Attributes

```
mysql> select * from attributes;
+-------+-----------------------------------------------------+-------+-----------------------------------------------------------------------+
| cfiID | cfiPath                                             | nfiID | nfiPath                                                               |
+-------+-----------------------------------------------------+-------+-----------------------------------------------------------------------+
|     1 | /i2nsf-cfi-policy/name                              |     1 | /i2nsf-security-policy/name                                           |
|     2 | /i2nsf-cfi-policy/language                          |     2 | /i2nsf-security-policy/language                                       |
|     3 | /i2nsf-cfi-policy/resolution-strategy               |     4 | /i2nsf-security-policy/resolution-strategy                            |
|     5 | /i2nsf-cfi-policy/rules/name                        |     7 | /i2nsf-security-policy/rules/name                                     |
|     6 | /i2nsf-cfi-policy/rules/priority                    |     9 | /i2nsf-security-policy/rules/priority                                 |
|     8 | /i2nsf-cfi-policy/rules/event/system-event          |    16 | /i2nsf-security-policy/rules/event/system-event                       |
|     9 | /i2nsf-cfi-policy/rules/event/system-alarm          |    17 | /i2nsf-security-policy/rules/event/system-alarm                       |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |    24 | /i2nsf-security-policy/rules/condition/layer-2/source-mac-address     |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |    49 | /i2nsf-security-policy/rules/condition/ipv4/source-ipv4-network       |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |    51 | /i2nsf-security-policy/rules/condition/ipv4/source-ipv4-range         |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |    71 | /i2nsf-security-policy/rules/condition/ipv6/source-ipv6-network       |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |    73 | /i2nsf-security-policy/rules/condition/ipv6/source-ipv6-range         |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |    81 | /i2nsf-security-policy/rules/condition/tcp/source-port-number         |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |   120 | /i2nsf-security-policy/rules/condition/udp/source-port-number         |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |   152 | /i2nsf-security-policy/rules/condition/sctp/source-port-number        |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |   185 | /i2nsf-security-policy/rules/condition/dccp/source-port-number        |
```

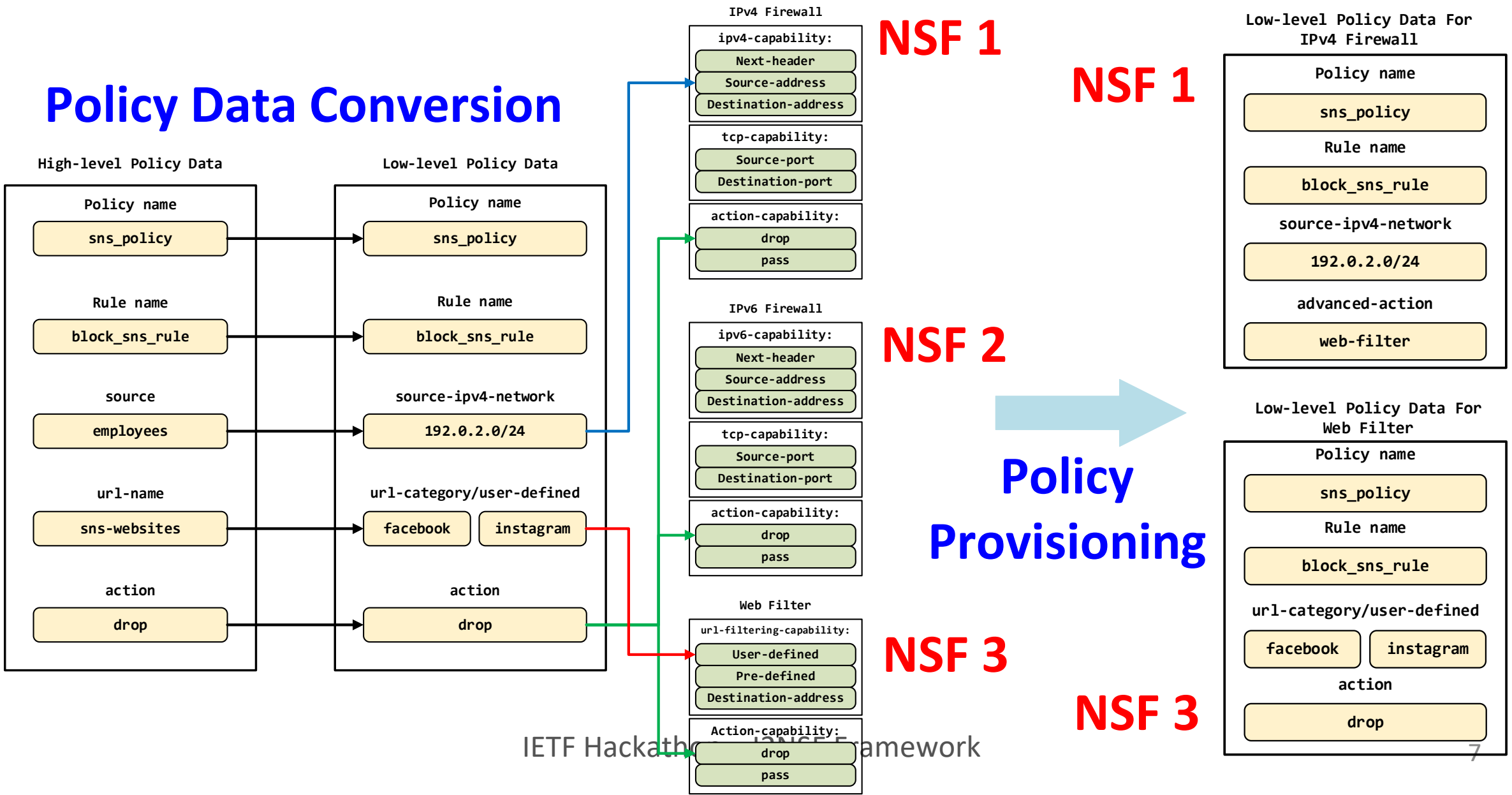
What got done (2/4)

High-level Security Policy

```
<i2nsf-cfi-policy
 xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-cfi-policy">
  <name>security_policy_for_blocking_sns</name>
  <rules>
    <name>block_access_to_sns_during_office_hours</name>
    <condition>
      <firewall>
        <source>employees</source>
      </firewall>
      <url>
        <url-name>sns-websites</url-name>
      </url>
    </condition>
    <actions>
      <primary-action>
        <action>drop</action>
      </primary-action>
    </actions>
  </rules>
</i2nsf-cfi-policy>
```

Extraction of High-Level Information

What got done (3/4): Policy Provisioning



What got done (4/4)

Generated Low-Level Policies

1. Low-Level Policy for Firewall

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
  <name>sns_access</name>
  <rules>
    <name>block_sns_access_during_operation_time_for_ipv4</name>
    <condition>
      <ipv4>
        <source-ipv4-network>192.0.2.0/24</source-ipv4-network>
      </ipv4>
    </condition>
    <action>
      <advanced-action>
        <content-security-control>
          url-filtering
        </content-security-control>
      </advanced-action>
    </action>
  </rules>
</i2nsf-security-policy>
```

employees translated to an IPv4 network address

2. Low-Level Policy for Web Filter

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
  <name>sns_access</name>
  <rules>
    <name>block_sns_access_during_operation_time</name>
    <condition>
      <url-category>
        <user-defined>Facebook</user-defined>
        <user-defined>Instagram</user-defined>
      </url-category>
    </condition>
    <action>
      <packet-action>
        <egress-action>drop</egress-action>
      </packet-action>
    </action>
  </rules>
</i2nsf-security-policy>
```

sns-websites translated into Facebook and Instagram

What we learn

- Automatic translation of a high-level security policy to a low-level security policy can be done.
- The automatic mapping between the two data models is done as there is similarity in “words” of labels in the attributes.
- The low-level policies can also be placed to the proper NSFs based on their capabilities through policy provisioning.

Next Step

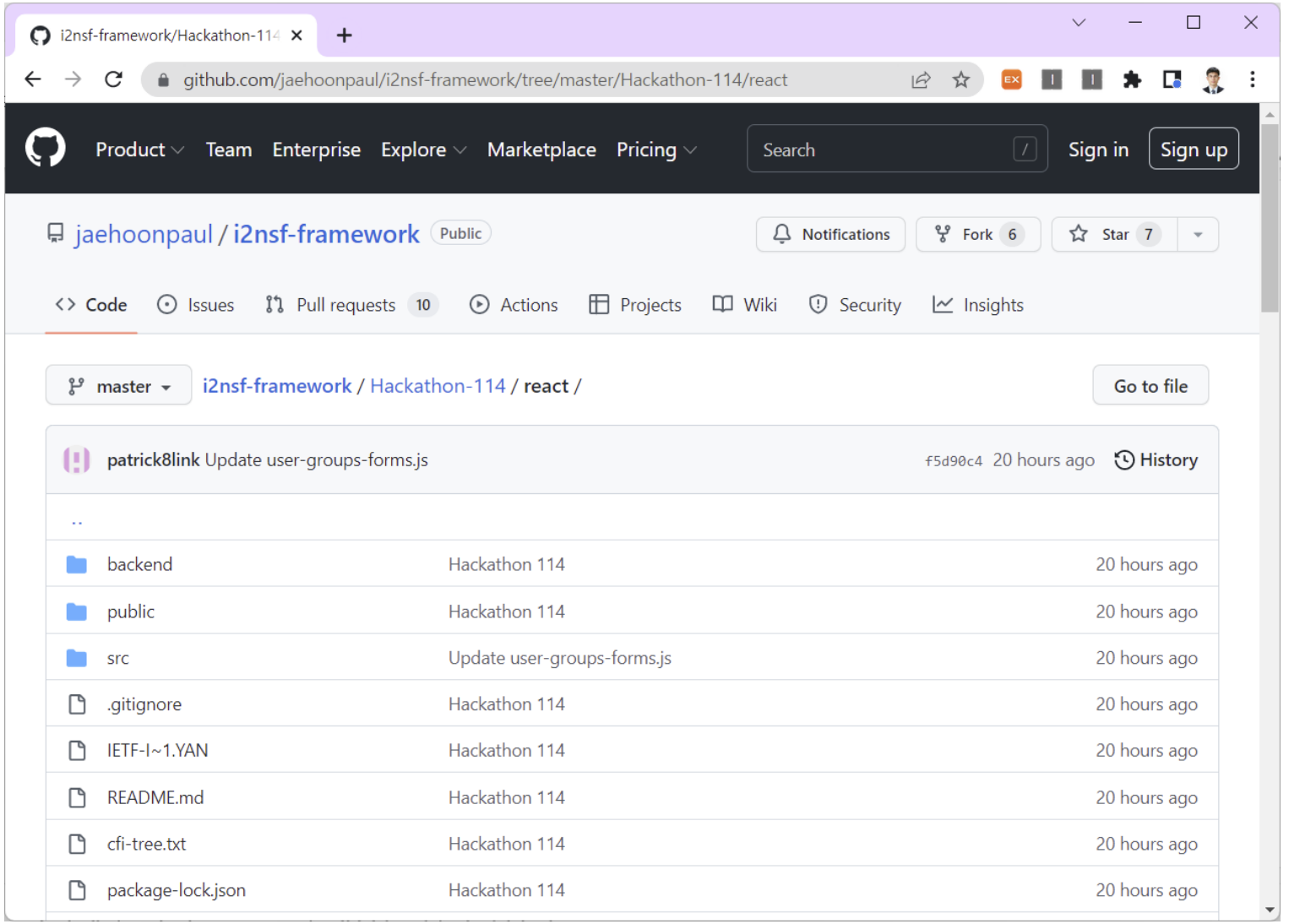
- Implementation of Intent-Based Networking (IBN) Security Provisioning through the integration of Security Policy Translator into Natural Language Processing (NLP), e.g., *block SNS websites from employees*.

- Implementation of Closed-Loop Security System for feedback and improvement of security policies based on monitoring data.

- Implementation of I2NSF Framework on top of Lightweight Cloud-Native System (e.g., Kubernetes).

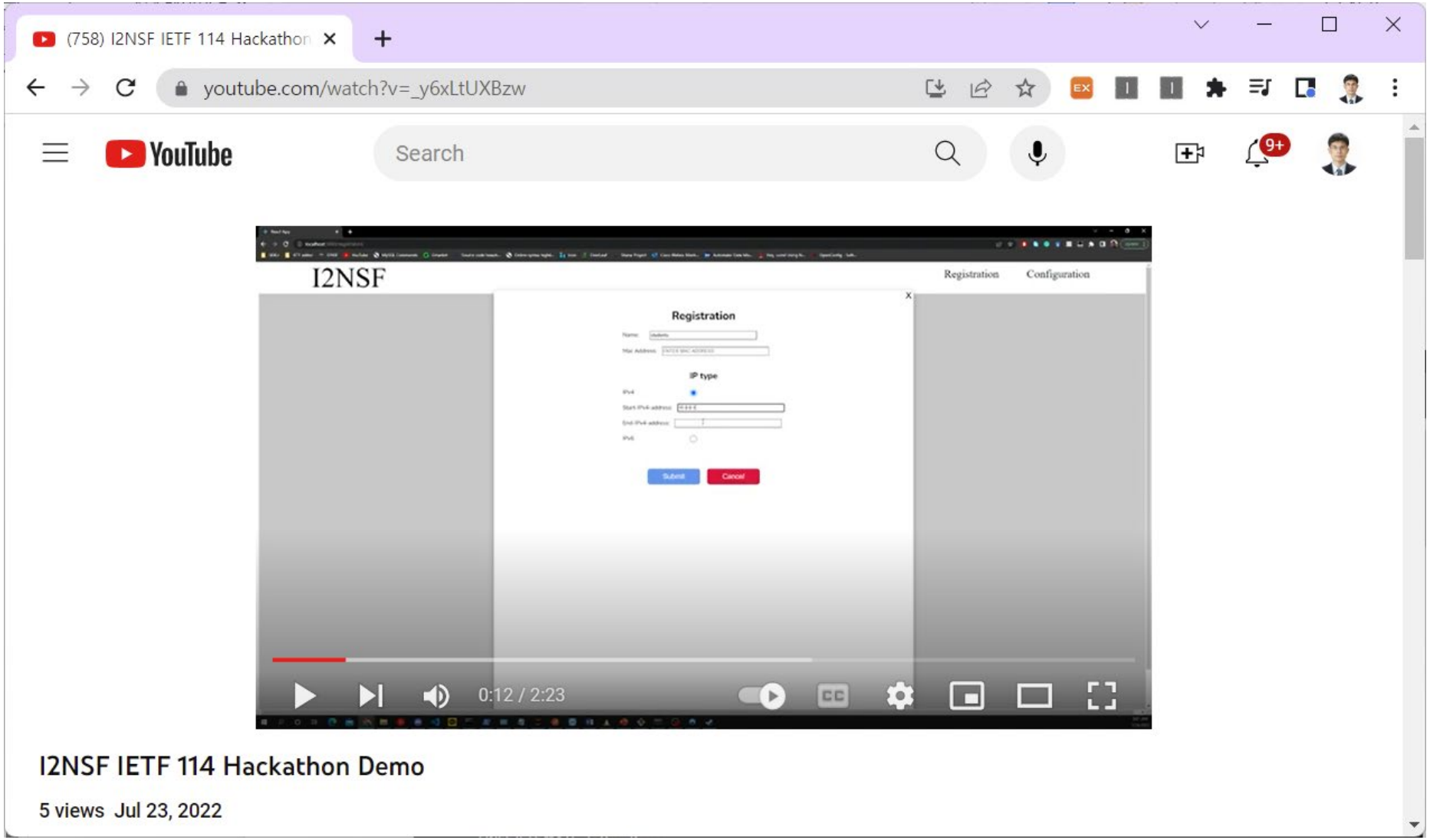
Open-Source Project at GitHub

URL: https://github.com/jaehoonpaul/i2nsf-framework



Demonstration Video Clip at YouTube

URL: https://youtu.be/_y6xLtUXBzw



Wrap Up

Hackathon Team

Champion:

- **Jaehoon Paul Jeong (SKKU)**

Professor:

- **Younghan Kim (SSU)**

Researchers:

- **Jung-Soo Park (ETRI)**
- **Yunchul Choi (ETRI)**
- **Jinyong Kim (SKKU)**

Students:

- **Patrick Lingga (SKKU)**
- **Jeonghyeon Kim (SKKU)**
- **Hadong Park (Calvin University)**

Hackathon Team Photo