IETF-112
I2NSF Hackathon Project

1-5 November 2021

Champion: Jaehoon (Paul) Jeong

Department of Computer Science and Engineering at SKKU

pauljeong@skku.edu

I2NSF (Interface to Network Security Functions) Framework Project

Champion: Jaehoon (Paul) Jeong

IETF-112 I2NSF Hackathon Project

Professors:

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

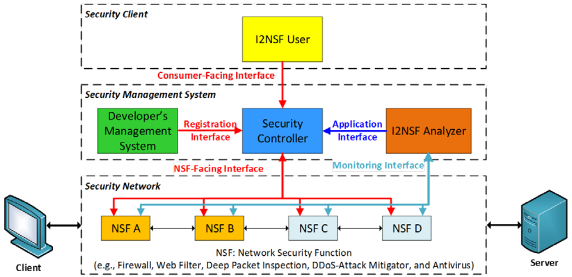
Researchers:

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

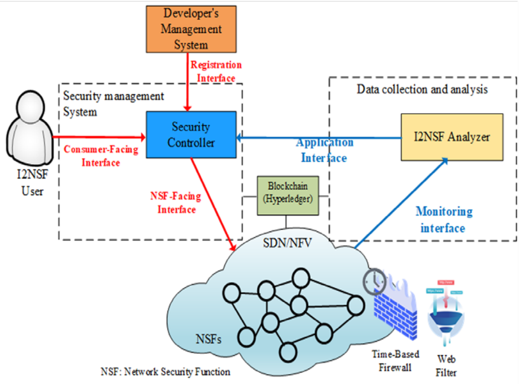
Students:

- Jeonghyeon Kim (SKKU)
- Patrick Lingga (SKKU)
- Kyungsik Kim (KNU)
- Cheolmin Kim (KNU)

I2NSF Framework


Security Client
I2NSF User
Consumer-Facing Interface
Security Management System
Developer's Management System
Registration Interface
Security Controller
Application Interface
I2NSF Analyzer
NSF-Facing Interface
Monitoring Interface
Security Network
NSF A
NSF B
NSF C
NSF D
Client
Server
NSF: Network Security Function
(e.g., Firewall, Web Filter, Deep Packet Inspection, DDoS-Attack Mitigator, and Antivirus)

I2NSF with Distributed Database System



Where to get Code and Demo Video Clip

- GitHub – Source Code
  - ✓ https://github.com/jaehoonpaul/i2nsf-framework
- YouTube – Demo Video Clip
  - ✓ https://youtu.be/dAA1WTGhlXE

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-IETF112-v1.md contain 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 with Automatic Data Model Mapper
  - ✓ Security Policy Provisioning
  - ✓ Distributed Database for NSF Monitoring Data

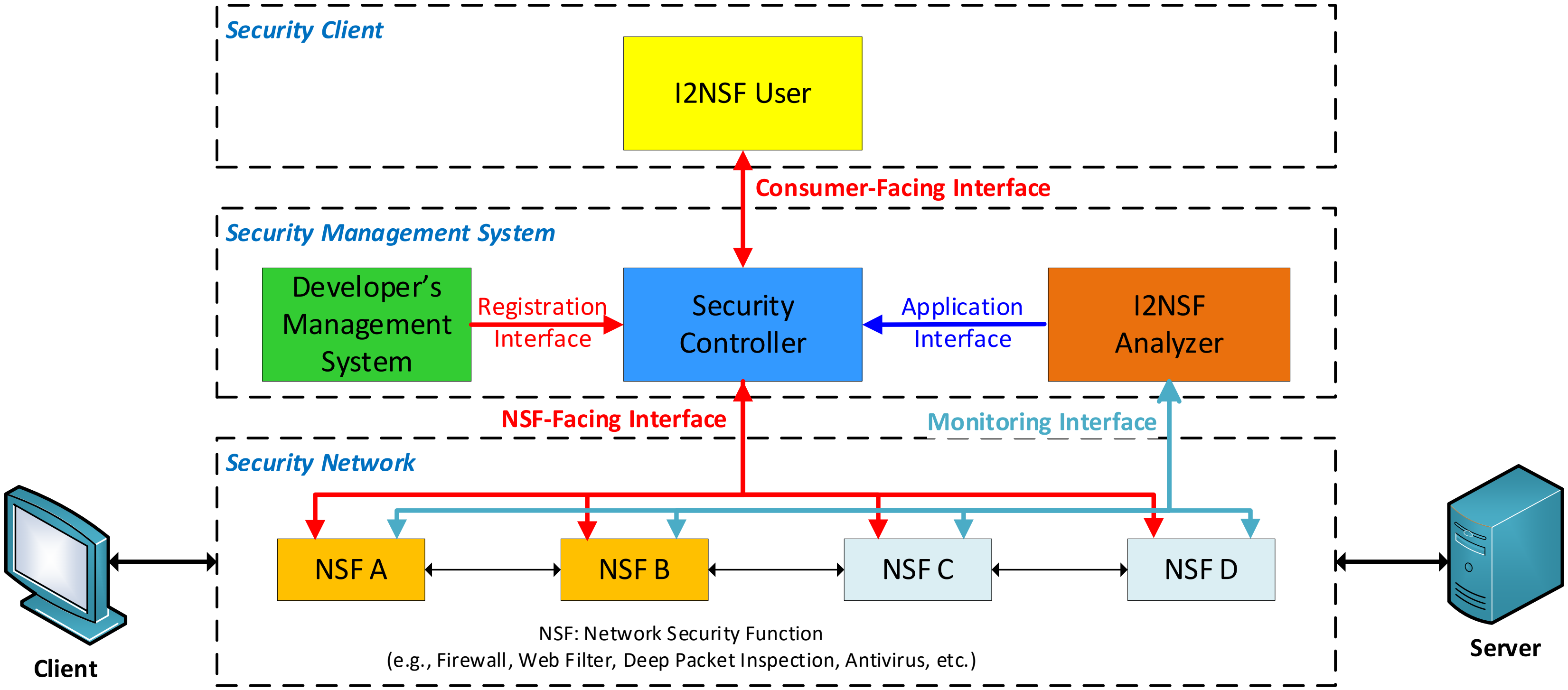
성균관대학교 SUNGKYUNKWAN UNIVERSITY　ETRI 한국전자통신연구원　openstack.　ConfD tail-f a Cisco company　SURICATA　HYPERLEDGER FABRIC

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-20
  - draft-ietf-i2nsf-consumer-facing-interface-dm-15
  - draft-ietf-i2nsf-nsf-facing-interface-dm-15
  - draft-ietf-i2nsf-registration-interface-dm-13
  - draft-ietf-i2nsf-nsf-monitoring-data-model-11
  - draft-yang-i2nsf-security-policy-translation-09
  - draft-jeong-i2nsf-security-management-automation-02

- ❖ Implementation of Distributed Network Auditing System for I2NSF Framework.
  - HyperLedger Fabric is used for Distributed Network Auditing System.

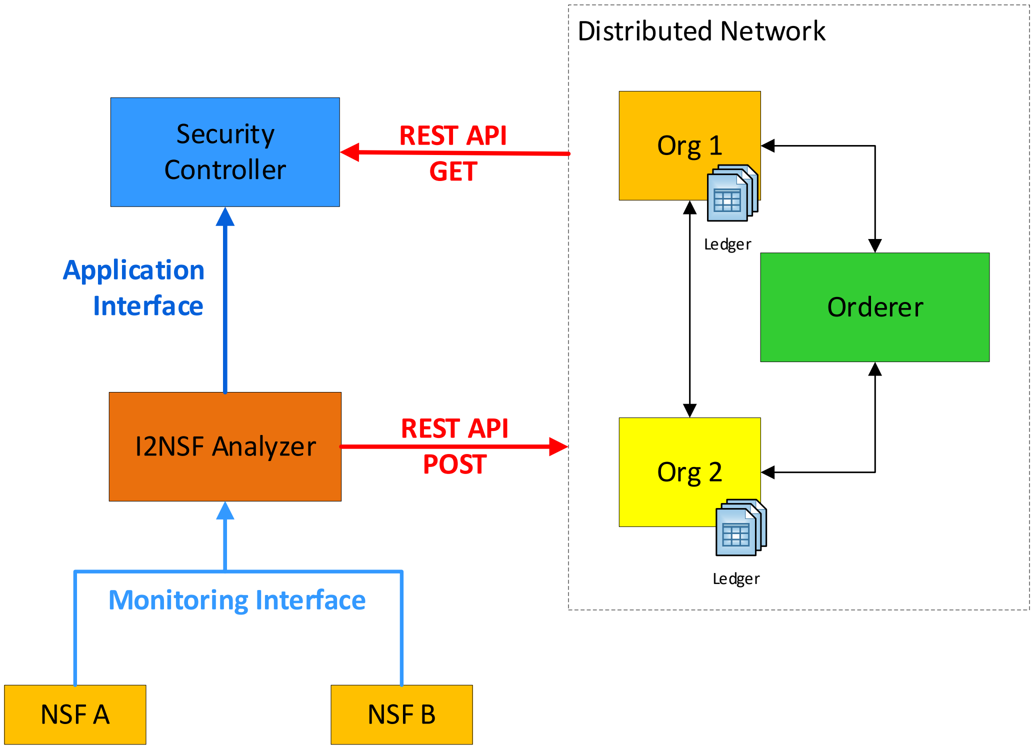
Hackathon Plan (2/2)



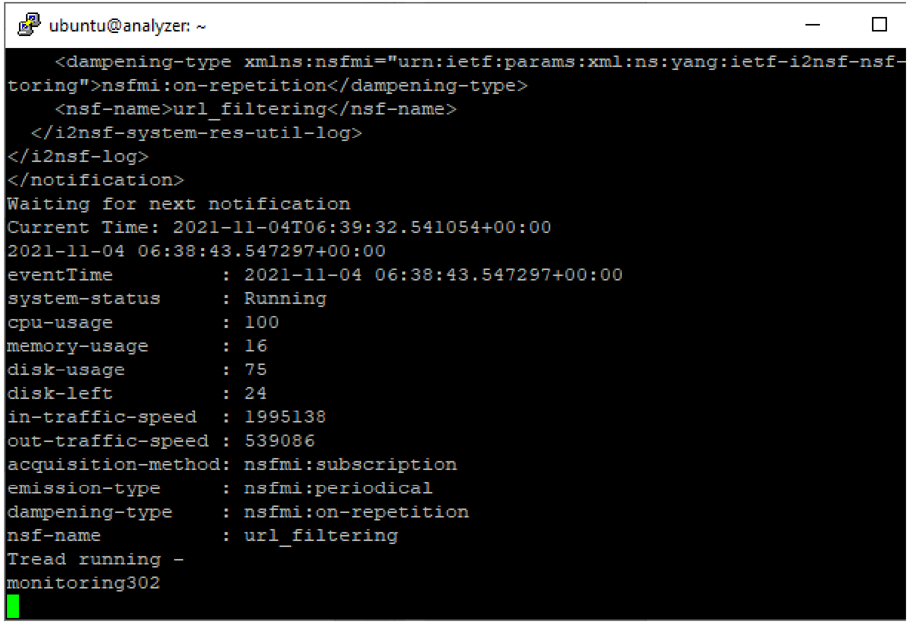
NSF: Network Security Function
(e.g., Firewall, Web Filter, Deep Packet Inspection, Antivirus, etc.)

What got done (1/3)

- Implementation of Distributed Database in I2NSF Framework

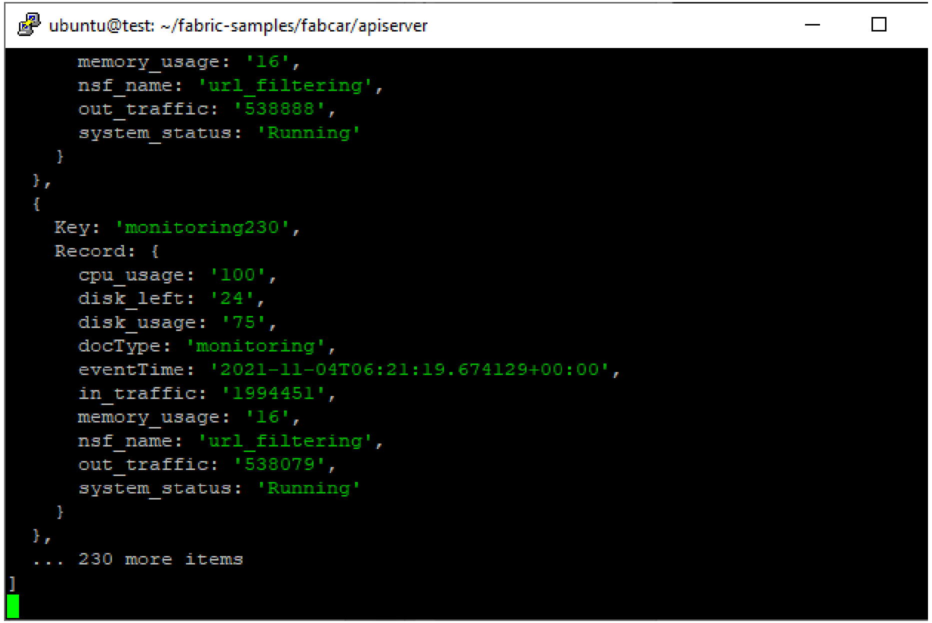


What got done (2/3)



```
    <dampening-type xmlns:nsfmi="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-
toring">nsfmi:on-repetition</dampening-type>
    <nsf-name>url_filtering</nsf-name>
  </i2nsf-system-res-util-log>
</i2nsf-log>
</notification>
Waiting for next notification
Current Time: 2021-11-04T06:39:32.541054+00:00
2021-11-04 06:38:43.547297+00:00
eventTime          : 2021-11-04 06:38:43.547297+00:00
system-status      : Running
cpu-usage          : 100
memory-usage       : 16
disk-usage         : 75
disk-left          : 24
in-traffic-speed   : 1995138
out-traffic-speed  : 539086
acquisition-method: nsfmi:subscription
emission-type      : nsfmi:periodical
dampening-type     : nsfmi:on-repetition
nsf-name           : url_filtering
Tread running -
monitoring302
```

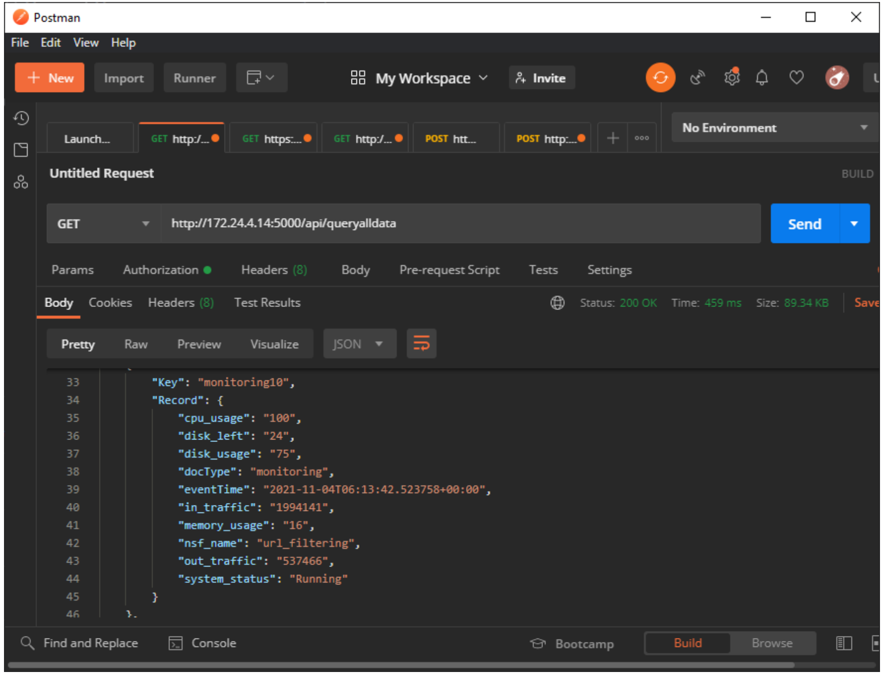
Push of NSF Monitoring Data to Distributed Database using REST API



```
      memory_usage: '16',
      nsf_name: 'url_filtering',
      out_traffic: '538888',
      system_status: 'Running'
    }
  },
  {
    Key: 'monitoring230',
    Record: {
      cpu_usage: '100',
      disk_left: '24',
      disk_usage: '75',
      docType: 'monitoring',
      eventTime: '2021-11-04T06:21:19.674129+00:00',
      in_traffic: '1994451',
      memory_usage: '16',
      nsf_name: 'url_filtering',
      out_traffic: '538079',
      system_status: 'Running'
    }
  },
  ... 230 more items
]
```

Distributed Database receives NSF Monitoring Data

What got done (3/3)



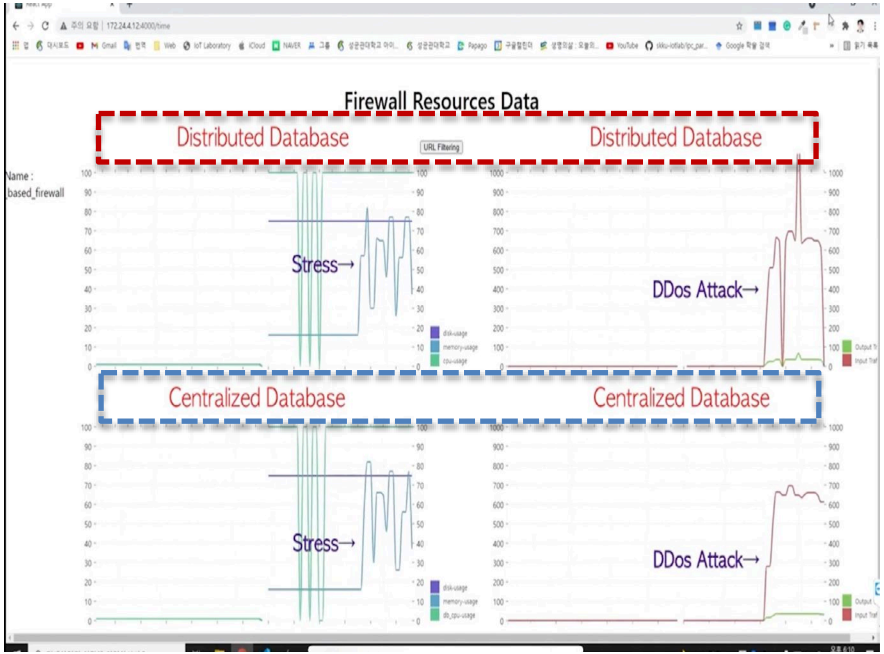
Untitled Request

GET http://172.24.4.14:5000/api/queryalldata Send

Status: 200 OK Time: 459 ms Size: 89.34 KB

```
            "Key": "monitoring10",
            "Record": {
                "cpu_usage": "100",
                "disk_left": "24",
                "disk_usage": "75",
                "docType": "monitoring",
                "eventTime": "2021-11-04T06:13:42.523758+00:00",
                "in_traffic": "1994141",
                "memory_usage": "16",
                "nsf_name": "url_filtering",
                "out_traffic": "537466",
                "system_status": "Running"
            }
        },
```

Pull of NSF Monitoring Data to Distributed Database using REST API



Real-time Visualization of Monitoring Data using Distributed Database

What we learn

- The usage of Distributed Database can tackle the possibility of data tampering in the I2NSF Framework.
- The distributed database system also denies the failure of a single point that is a major problem of a centralized database.
- Overall, the distributed database system can improve the security and reliability of the I2NSF Framework.

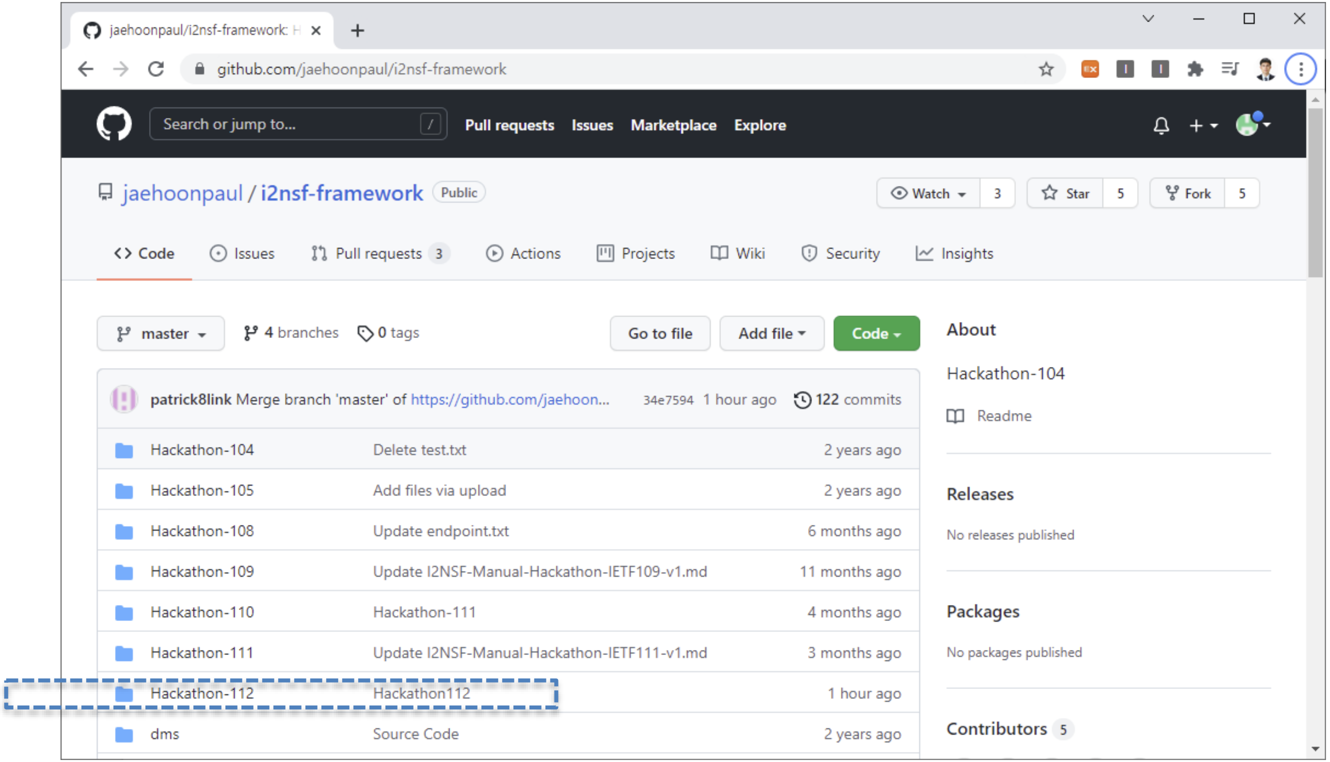
Next Step

- Current implementation only stores NSF monitoring data into the distributed database system.

- As future work, the whole I2NSF data and information (e.g., NSF monitoring data, security policy, and NSF capabilities) will be stored into the distributed database system.

Open-Source Project at GitHub

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



jaehoonpaul/i2nsf-framework: H
github.com/jaehoonpaul/i2nsf-framework
Search or jump to...
Pull requests
Issues
Marketplace
Explore

jaehoonpaul / i2nsf-framework Public

Watch 3 | Star 5 | Fork 5

4 branches · 0 tags

patrick8link Merge branch 'master' of https://github.com/jaehoon... 34e7594 1 hour ago · 122 commits

Hackathon-104	Delete test.txt	2 years ago
Hackathon-105	Add files via upload	2 years ago
Hackathon-108	Update endpoint.txt	6 months ago
Hackathon-109	Update I2NSF-Manual-Hackathon-IETF109-v1.md	11 months ago
Hackathon-110	Hackathon-111	4 months ago
Hackathon-111	Update I2NSF-Manual-Hackathon-IETF111-v1.md	3 months ago
Hackathon-112	Hackathon112	1 hour ago
dms	Source Code	2 years ago

About

Hackathon-104

Readme

Releases

No releases published

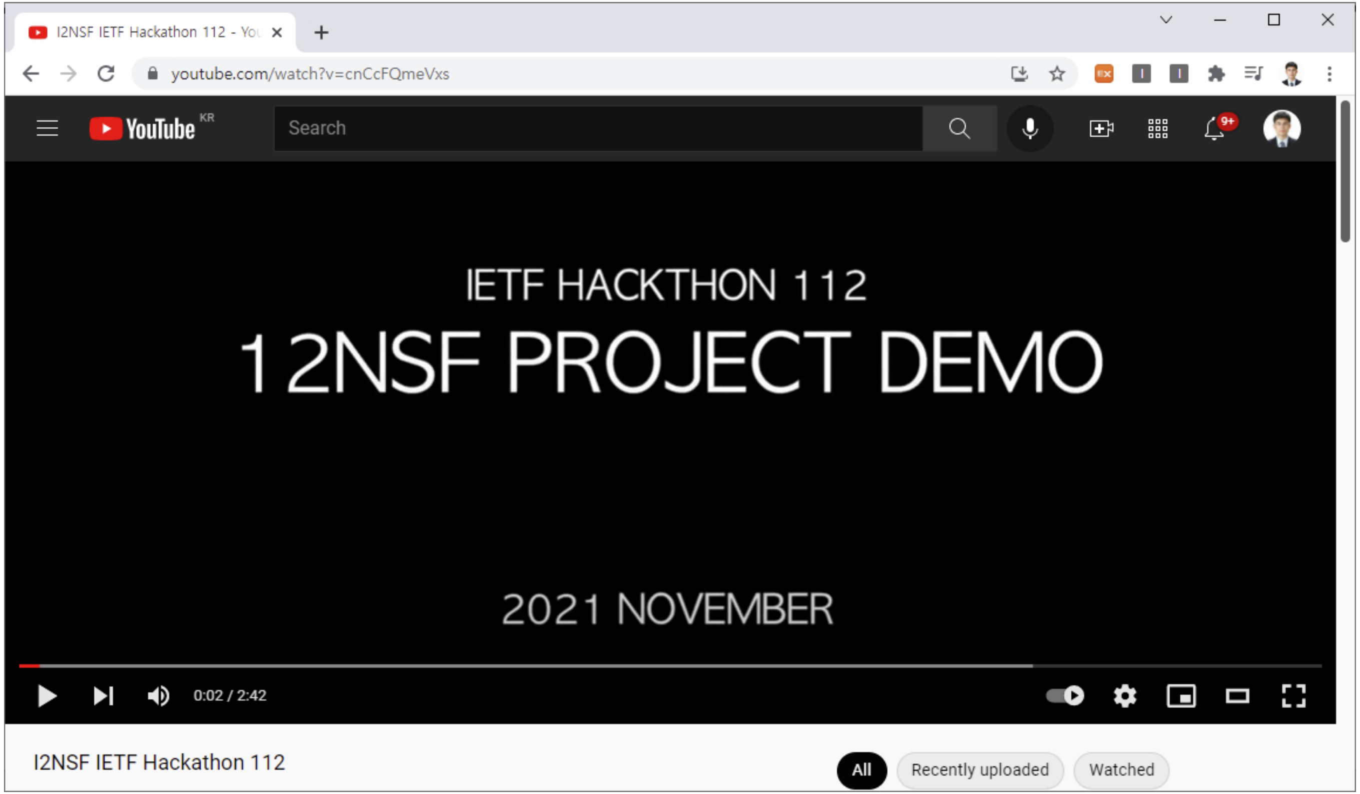
Packages

No packages published

Contributors 5

Demonstration Video Clip at YouTube

URL: https://www.youtube.com/watch?v=cnCcFQmeVxs



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)
- Cheolmin Kim (KNU)
- Kyungsik Kim (KNU)

Hackathon Team Photo

IETF-112 Hackathon Korea Teams

Sponsors

NRF National Research Foundation of Korea

IITP Institute of Information & Communications Technology Planning & Evaluation

TTA 한국정보통신기술협회 Telecommunications Technology Association

SUNGKYUNKWAN UNIVERSITY 1398 성균관대학교

숭실대학교 Soongsil University

ETRI 한국전자통신연구원 Electronics and Telecommunications Research Institute

K-IETF KOREA IETF FORUM

Appendix (1/2)

- The distributed database network is implemented using Hyperledger Fabric version 2.2.

- The setup of Hyperledger can be done by following the steps in https://hyperledger-fabric.readthedocs.io/en/release-2.2/test_network.html

- To configure the data model for the Monitoring Interface YANG Data Model, edit the chaincode with the Monitoring Interface YANG Data Model. See in our GitHub chaincode files.

- In our case, JavaScript is used to edit Hyperledger functions (e.g., initLedger, query, create, and queryAll) for the Monitoring Interface YANG Data Model.

Appendix (2/2)

- To save the monitoring data, we implemented REST API. Run apiserver.js to execute a REST server.

- The URLs for I2NSF User to GET (pull) data using REST API are:
  - http://ip-address:5000/api/queryalldata
  - http://ip-address:5000/api/querylastdata

- The URLs for the I2NSF Analyzer to POST (push) data using REST API are:
  - http://ip-address:5000/api/adddata

❖ Note: ip-address means the specific IP address of the distributed database.