IETF-124 Hackathon

5G-I2NSF: An Integrated Security System for 5G Networks with I2NSF

November 2, 2025, Montreal

Champion: Jaehoon (Paul) Jeong

Members: Joseph Ahn, Jiwon Suh, and Jiwon Yoo

Department of Computer Science and Engineering at SKKU

Email: {pauljeong, ahnjs124, sjw6136, sowra1}@skku.edu

IETF-124 5G-I2NSF System for Integrated Security Services in 5G Networks

Champion: Jaehoon Paul Jeong (SKKU)

IETF-124 5G-I2NSF Project

Professors:

- Jaehoon Paul Jeong (SKKU)
- Younghan Kim (SSU)
- Yong-Geun Hong (DJU)
- Joo-Sang Youn (DEU)

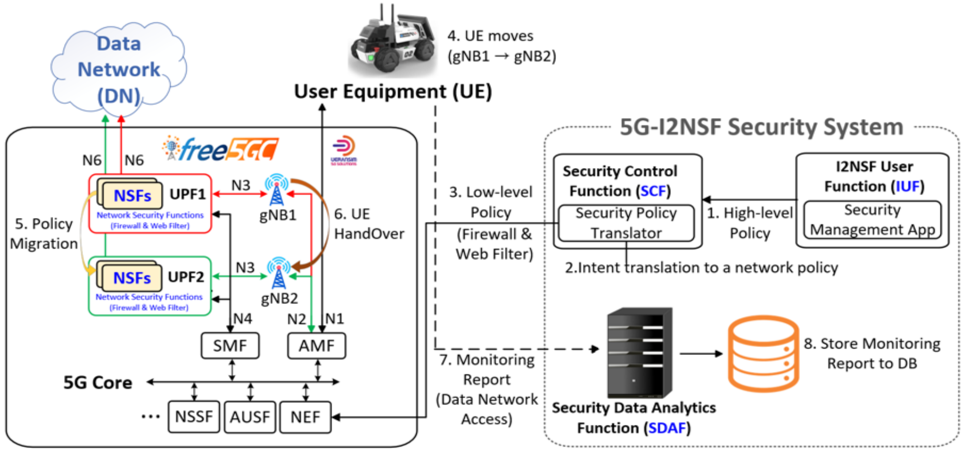
Researchers:

- Jung-Soo Park (ETRI)
- Yunchul Choi (ETRI)

Students:

- Yoseop Ahn (SKKU)
- Jiwon Suh (SKKU)
- Jiwon Yoo (SKKU)

5G-I2NSF System


4. UE moves (gNB1 → gNB2)
Data Network (DN)
User Equipment (UE)
N6
N6
free5GC
NSFs
UPF1
Network Security Functions (Firewall & Web Filter)
N3
gNB1
5. Policy Migration
6. UE HandOver
NSFs
UPF2
Network Security Functions (Firewall & Web Filter)
N3
gNB2
N4
N2
N1
SMF
AMF
5G Core
…
NSSF
AUSF
NEF
3. Low-level Policy (Firewall & Web Filter)
7. Monitoring Report (Data Network Access)
5G-I2NSF Security System
Security Control Function (SCF)
Security Policy Translator
1. High-level Policy
I2NSF User Function (IUF)
Security Management App
2.Intent translation to a network policy
Security Data Analytics Function (SDAF)
8. Store Monitoring Report to DB

Objectives

- This study presents an integrated framework for automated security orchestration in 5G edge networks based on the Interface to Network Security Functions (I2NSF) system.
- The proposed system translates a high-level security policy in YANG/XML into the corresponding low-level security policy in YANG/XML and dynamically deploys the corresponding Network Security Functions (NSFs) within the User Plane Function (UPF).
- It also supports seamless policy migration during UE handovers, minimizing latency and ensuring consistent security enforcement across distributed edge environments.

Future Work

- We plan to leverage LLMs to interpret natural language intents and generate YAML-based policies automatically, while AI-driven analysis will enhance security through adaptive learning, anomaly detection, and policy optimization.

5G-I2NSF Development Environment

- OS: Ubuntu 22.04
- Free5GC VM: version 4.1.0
- UERANSIM VM (UE & RAN): version 3.2.6
- GitHub Repository: https://github.com/jaehoonpauljeong/5G-I2NSF

Workflow of the 5G-I2NSF Testbed

1. A high-level security policy is generated by the I2NSF User Function (IUF) based on a user's intent.
2. The Security Control Function (SCF) translates this high-level policy into the corresponding low-level policy.
3. The low-level policy is then delivered to the relevant 5G Core functions AMF and SMF via NEF, and appropriate Network Security Functions (NSFs) are instantiated within a UPF according to this policy.
4. After the NSFs are deployed within a UPF, the UE connected to gNB1 moves to gNB2 by handover.
5. By predicting the next gNB to which the UE will move, the Security Policy for NSFs (e.g., Firewall & Web Filter) in UPF1 in gNB1 is proactively migrated to UPF2 in gNB2.
6. Following the policy migration, the AMF manages the UE's handover procedure, while the SMF reestablishes the N4 session with UPF2.
7. The UE's data network access logs are collected and sent to the Security Data Analytics Function (SDAF) as monitoring reports.
8. These reports are analyzed by SDAF to verify whether the security policy is enforced well by NSFs or not.

Goal of Hackathon Project

- **Goal**
  - To make an Integrated Security System in 5G Networks with Interface to Network Security Functions (I2NSF) Framework.

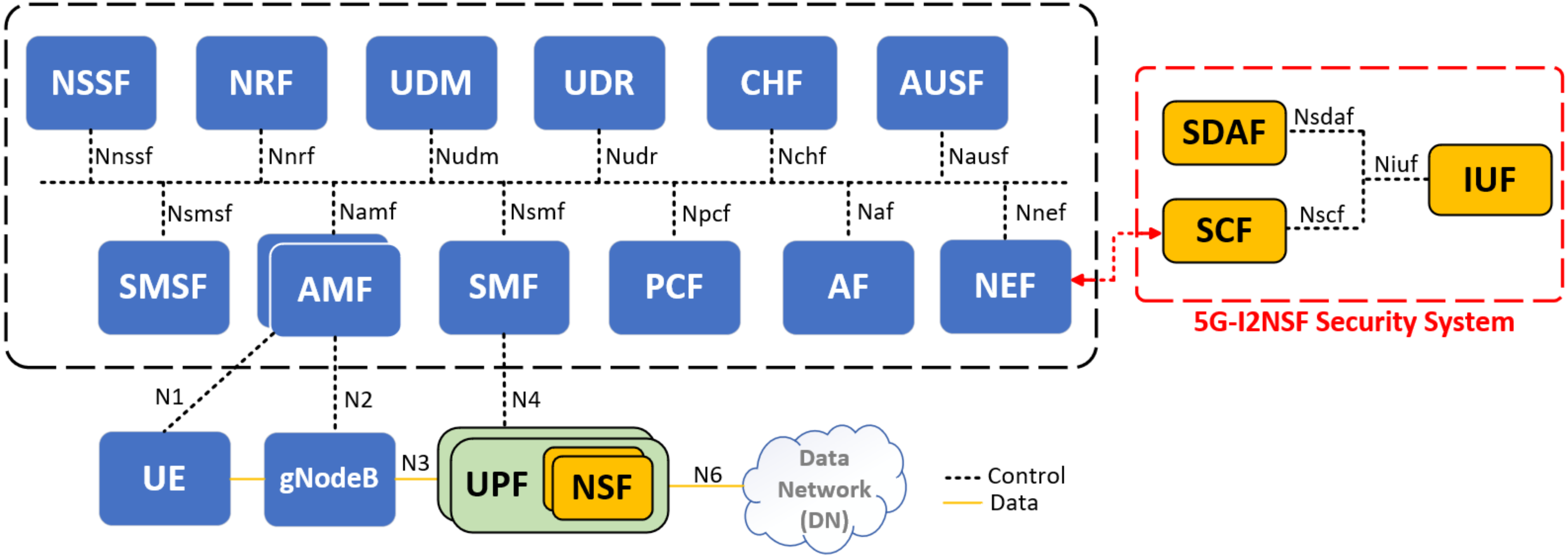
- **Security Policy Provisioning** through **Edge-Based I2NSF Framework**
  - Integration of I2NSF to 5G as Edge Approach rather than Cloud Approach.
  - Formation of I2NSF Components as 5G Network Functions (NFs) and also I2NSF Interfaces as 5G Interfaces.

- **Internet Drafts for the 5G-I2NSF Project**
  - https://datatracker.ietf.org/doc/draft-ahn-nmrg-5g-security-i2nsf-framework/
  - https://datatracker.ietf.org/doc/draft-gu-nmrg-intent-translator/

I2NSF-based Integration for 5G Security Services

➢ Integration of I2NSF-based Components in 5G Service-Based Architecture

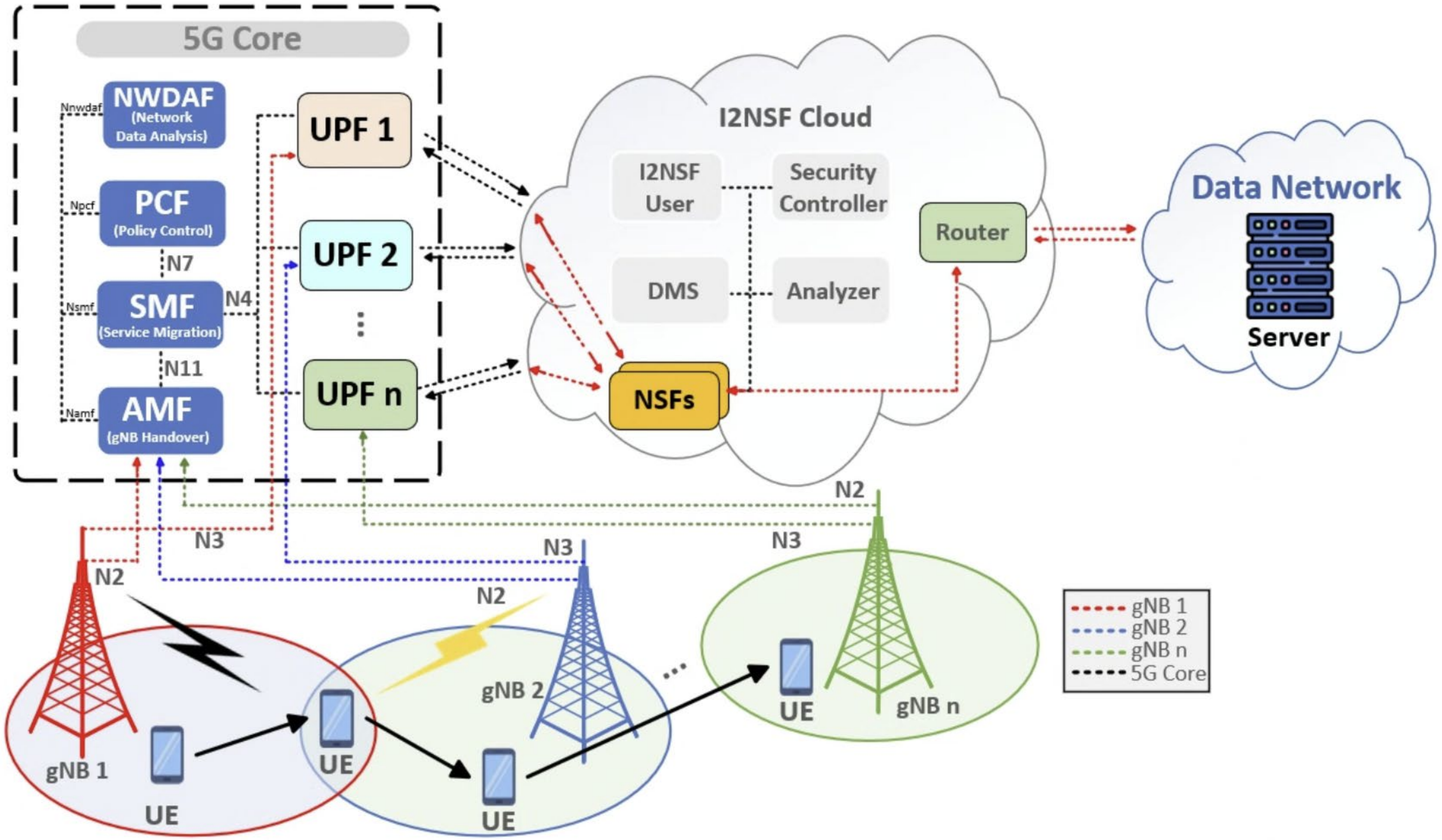
- There are New Components defined for I2NSF for 5G such as the I2NSF User Function (IUF), Security Control Function (SCF), Security Data Analytics Function (SDAF), Developer's Management Functions (DMF) and Network Security Functions (NSF).
- These are integrated to 5G Core Networks for efficient security policy provisioning.



An Integrated Security System for 5G Networks with I2NSF

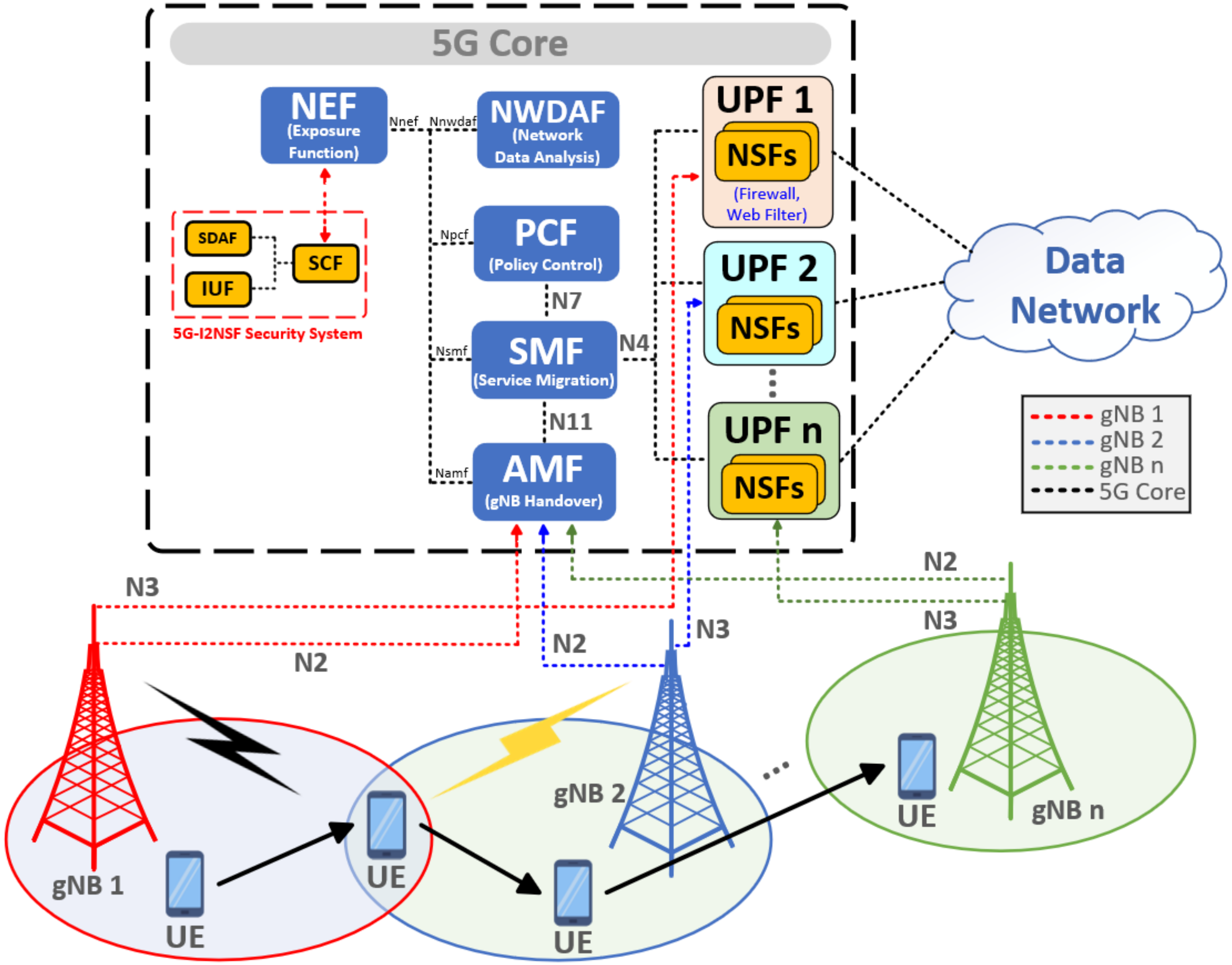
I2NSF Cloud Approach for 5G Networks

- **Legacy: Execution of NSFs in I2NSF Cloud outside of 5G Core Networks**
  - **Long delay by a detoured path from a UE to I2NSF Cloud for security services.**



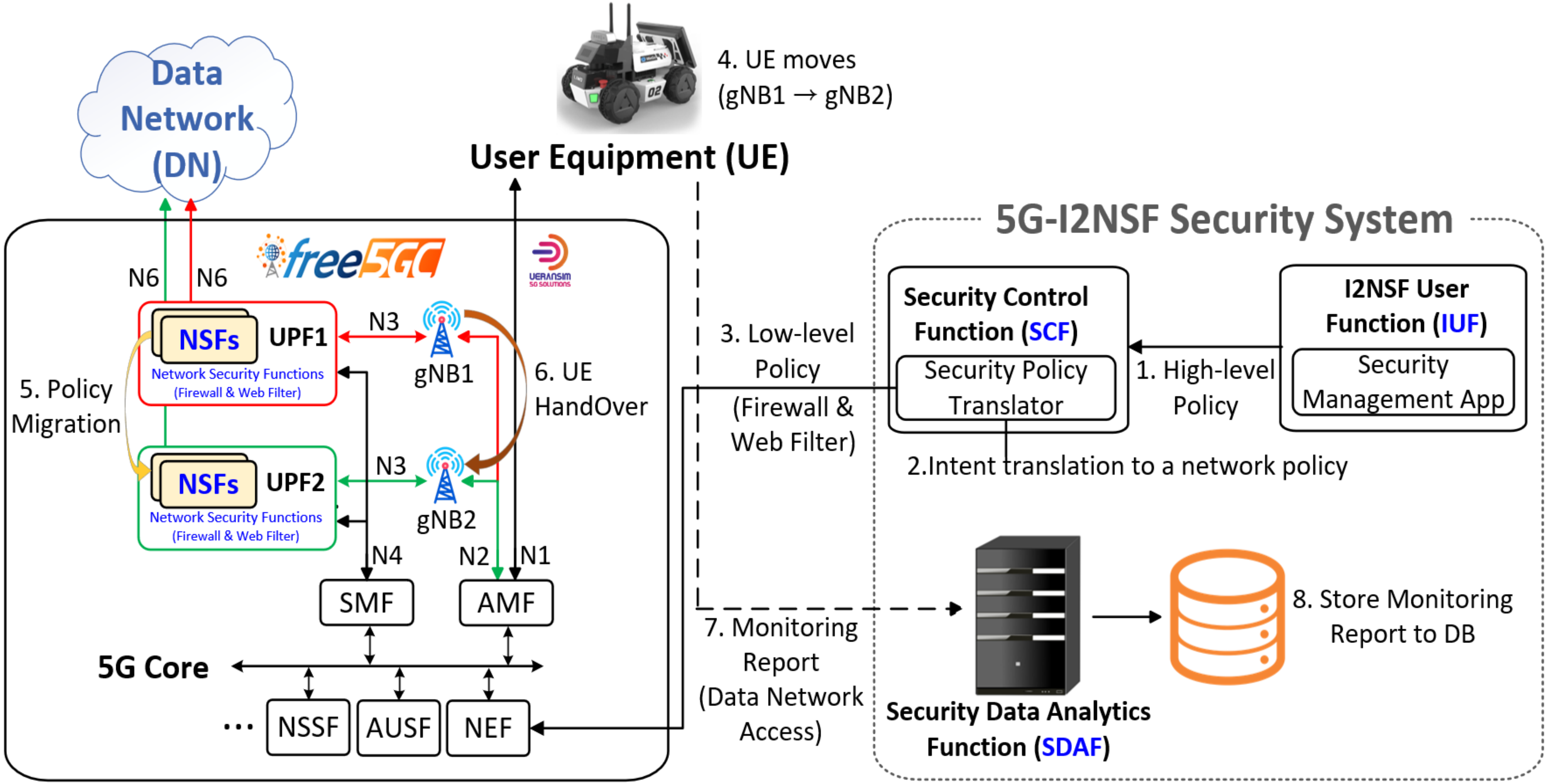
I2NSF Edge Approach for 5G Networks

- **Legacy:** Execution of NSFs in I2NSF Cloud inside of 5G Core Networks
  - Short delay by an optimal path from a UE to I2NSF Edge for security services.



An Integrated Security System for 5G Networks with I2NSF

➢ 5G-I2NSF System



Data Network (DN)
4. UE moves (gNB1 → gNB2)
User Equipment (UE)
free5GC
UERANSIM 5G SOLUTIONS
N6
N6
NSFs
UPF1
Network Security Functions (Firewall & Web Filter)
N3
gNB1
5. Policy Migration
6. UE HandOver
NSFs
UPF2
Network Security Functions (Firewall & Web Filter)
N3
gNB2
N4
N2
N1
SMF
AMF
5G Core
…
NSSF
AUSF
NEF
3. Low-level Policy (Firewall & Web Filter)
7. Monitoring Report (Data Network Access)
5G-I2NSF Security System
Security Control Function (SCF)
Security Policy Translator
1. High-level Policy
I2NSF User Function (IUF)
Security Management App
2.Intent translation to a network policy
8. Store Monitoring Report to DB
Security Data Analytics Function (SDAF)

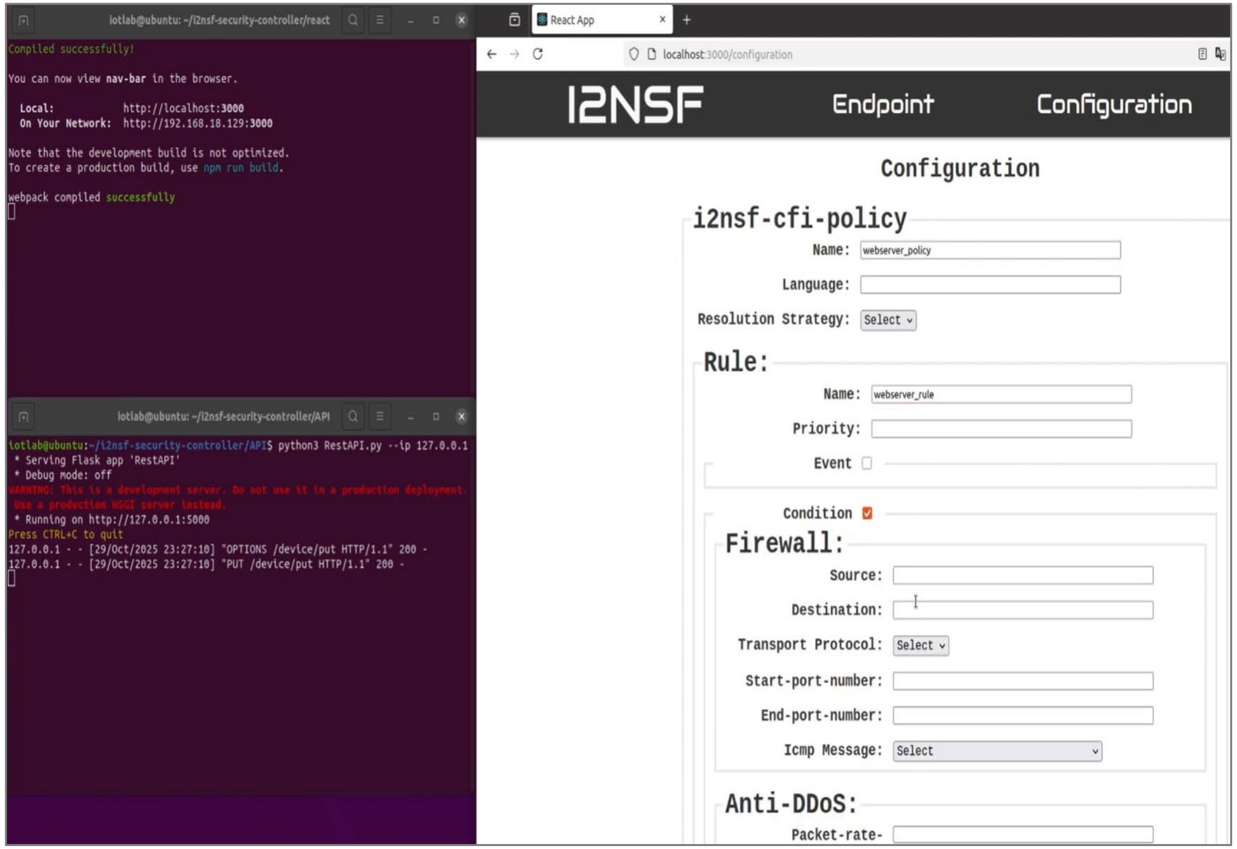
What we learned

- We learned the design and implementation of I2NSF for 5G Networks with Free5GC.

- We proved the effectiveness and efficiency of the integrated security system with 5G-I2NSF.

Demonstration of 5G-I2NSF

➢ Security Policy Translation


iotlab@ubuntu: ~/i2nsf-security-controller/react
React App
localhost:3000/configuration
I2NSF
Endpoint
Configuration

```
Compiled successfully!

You can now view nav-bar in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.18.129:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

iotlab@ubuntu: ~/i2nsf-security-controller/API

```
iotlab@ubuntu:~/i2nsf-security-controller/API$ python3 RestAPI.py --ip 127.0.0.1
 * Serving Flask app 'RestAPI'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
 Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [29/Oct/2025 23:27:10] "OPTIONS /device/put HTTP/1.1" 200 -
127.0.0.1 - - [29/Oct/2025 23:27:10] "PUT /device/put HTTP/1.1" 200 -
```

Configuration

i2nsf-cfi-policy

Name: webserver_policy

Language:

Resolution Strategy: Select

Rule:

Name: webserver_rule

Priority:

Event □

Condition ☑

Firewall:

Source:

Destination:

Transport Protocol: Select

Start-port-number:

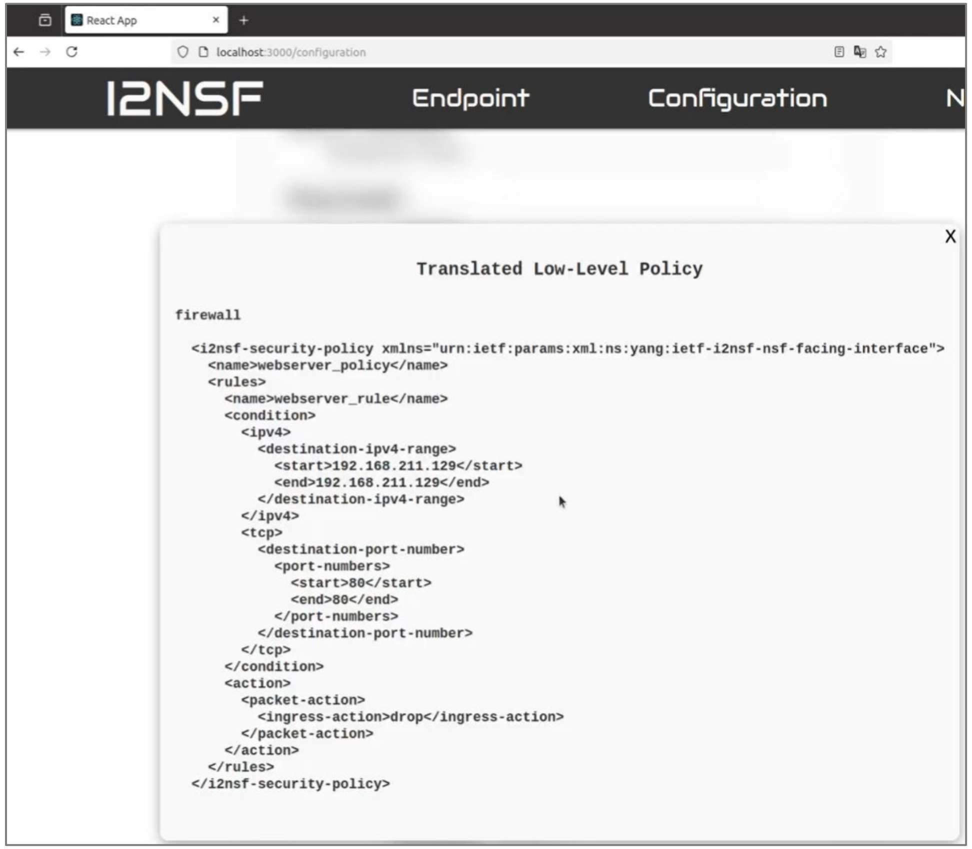
End-port-number:

Icmp Message: Select

Anti-DDoS:

Packet-rate-

Web-Based I2NSF User Function for High-level Security Policy



Translated Low-Level Policy

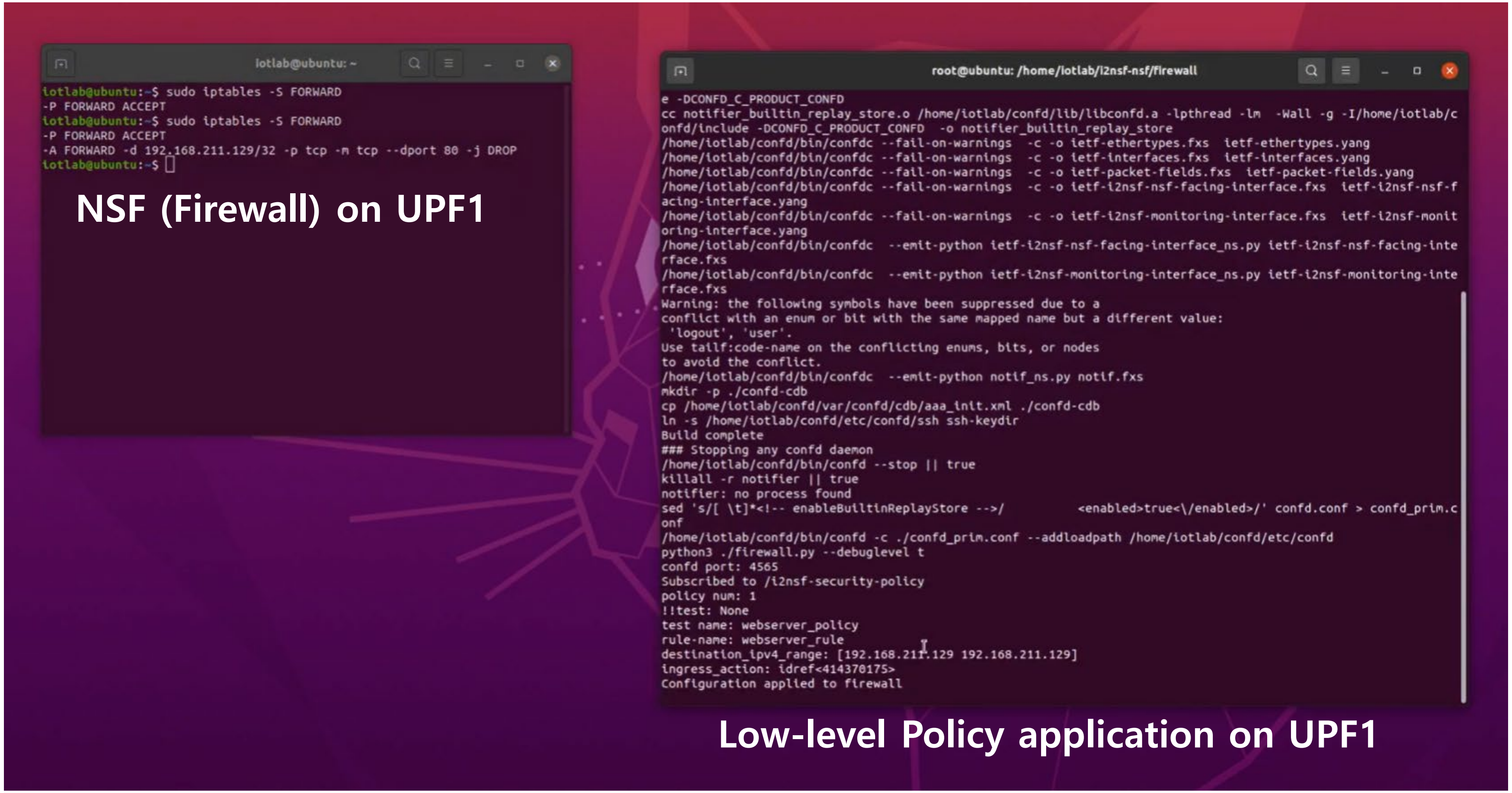
```
firewall

  <i2nsf-security-policy xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
    <name>webserver_policy</name>
    <rules>
      <name>webserver_rule</name>
      <condition>
        <ipv4>
          <destination-ipv4-range>
            <start>192.168.211.129</start>
            <end>192.168.211.129</end>
          </destination-ipv4-range>
        </ipv4>
        <tcp>
          <destination-port-number>
            <port-numbers>
              <start>80</start>
              <end>80</end>
            </port-numbers>
          </destination-port-number>
        </tcp>
      </condition>
      <action>
        <packet-action>
          <ingress-action>drop</ingress-action>
        </packet-action>
      </action>
    </rules>
  </i2nsf-security-policy>
```

Low-level Security Policy Generation by Security Policy Translator

Demonstration of 5G-I2NSF

➢ NSF (Firewall) generation on UPF



```
iotlab@ubuntu:~$ sudo iptables -S FORWARD
-P FORWARD ACCEPT
iotlab@ubuntu:~$ sudo iptables -S FORWARD
-P FORWARD ACCEPT
-A FORWARD -d 192.168.211.129/32 -p tcp -m tcp --dport 80 -j DROP
iotlab@ubuntu:~$ 
```

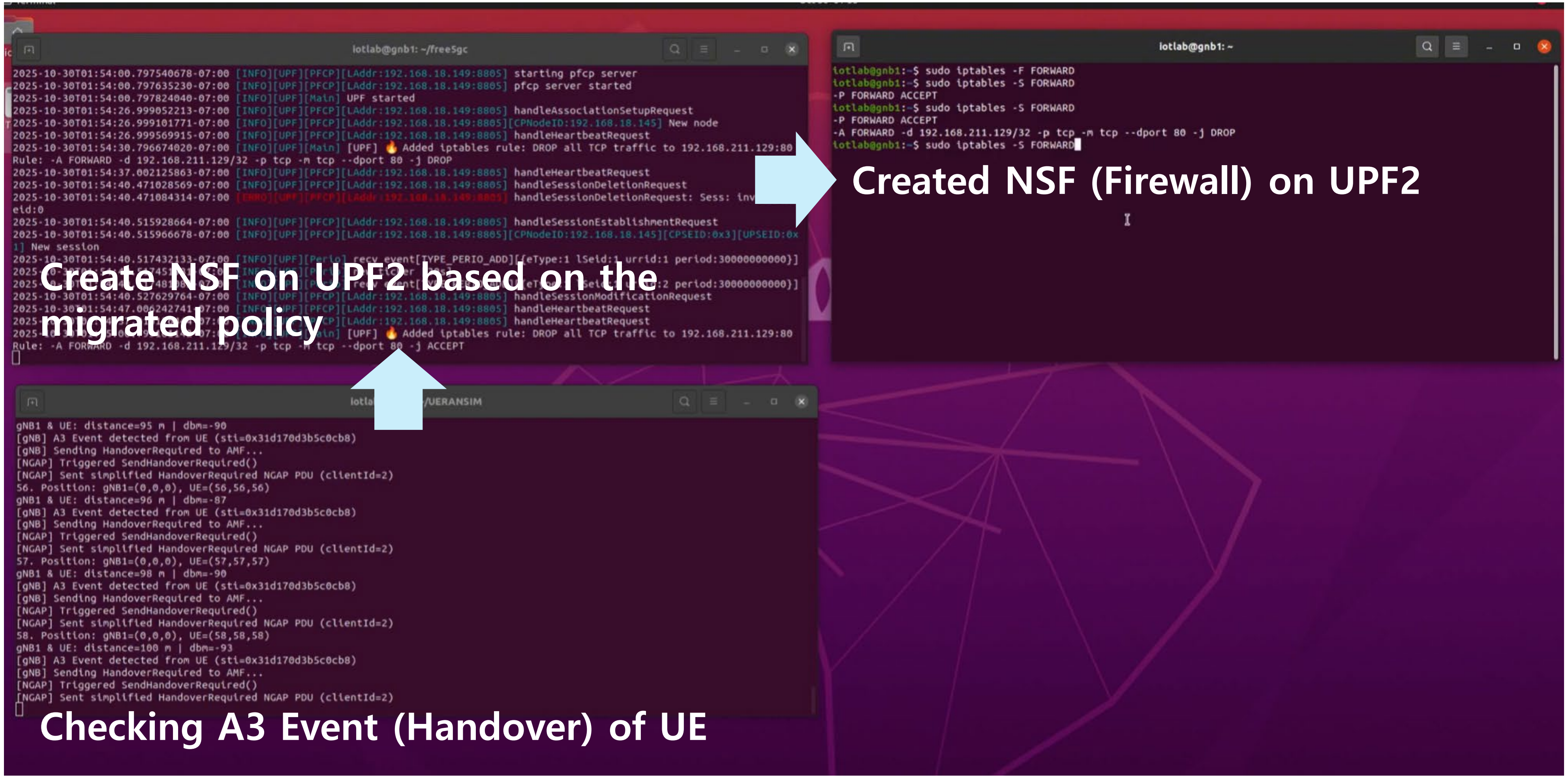
NSF (Firewall) on UPF1

```
e -DCONFD_C_PRODUCT_CONFD
cc notifier_builtin_replay_store.o /home/iotlab/confd/lib/libconfd.a -lpthread -lm  -Wall -g -I/home/iotlab/c
onfd/include -DCONFD_C_PRODUCT_CONFD  -o notifier_builtin_replay_store
/home/iotlab/confd/bin/confdc --fail-on-warnings  -c -o ietf-ethertypes.fxs  ietf-ethertypes.yang
/home/iotlab/confd/bin/confdc --fail-on-warnings  -c -o ietf-interfaces.fxs  ietf-interfaces.yang
/home/iotlab/confd/bin/confdc --fail-on-warnings  -c -o ietf-packet-fields.fxs  ietf-packet-fields.yang
/home/iotlab/confd/bin/confdc --fail-on-warnings  -c -o ietf-i2nsf-nsf-facing-interface.fxs  ietf-i2nsf-nsf-f
acing-interface.yang
/home/iotlab/confd/bin/confdc --fail-on-warnings  -c -o ietf-i2nsf-monitoring-interface.fxs  ietf-i2nsf-monit
oring-interface.yang
/home/iotlab/confd/bin/confdc  --emit-python ietf-i2nsf-nsf-facing-interface_ns.py ietf-i2nsf-nsf-facing-inte
rface.fxs
/home/iotlab/confd/bin/confdc  --emit-python ietf-i2nsf-monitoring-interface_ns.py ietf-i2nsf-monitoring-inte
rface.fxs
Warning: the following symbols have been suppressed due to a
conflict with an enum or bit with the same mapped name but a different value:
 'logout', 'user'.
Use tailf:code-name on the conflicting enums, bits, or nodes
to avoid the conflict.
/home/iotlab/confd/bin/confdc  --emit-python notif_ns.py notif.fxs
mkdir -p ./confd-cdb
cp /home/iotlab/confd/var/confd/cdb/aaa_init.xml ./confd-cdb
ln -s /home/iotlab/confd/etc/confd/ssh ssh-keydir
Build complete
### Stopping any confd daemon
/home/iotlab/confd/bin/confd --stop || true
killall -r notifier || true
notifier: no process found
sed 's/[ \t]*<!-- enableBuiltinReplayStore -->/          <enabled>true<\/enabled>/' confd.conf > confd_prim.c
onf
/home/iotlab/confd/bin/confd -c ./confd_prim.conf --addloadpath /home/iotlab/confd/etc/confd
python3 ./firewall.py --debuglevel t
confd port: 4565
Subscribed to /i2nsf-security-policy
policy num: 1
!!test: None
test name: webserver_policy
rule-name: webserver_rule
destination_ipv4_range: [192.168.211.129 192.168.211.129]
ingress_action: idref<414370175>
Configuration applied to firewall
```

Low-level Policy application on UPF1

Demonstration of 5G-I2NSF

- UE handover and policy-based NSF creation on UPF



```
2025-10-30T01:54:00.797540678-07:00 [INFO][UPF][PFCP][LAddr:192.168.18.149:8805] starting pfcp server
2025-10-30T01:54:00.797635230-07:00 [INFO][UPF][PFCP][LAddr:192.168.18.149:8805] pfcp server started
2025-10-30T01:54:00.797824040-07:00 [INFO][UPF][Main] UPF started
2025-10-30T01:54:26.999052213-07:00 [INFO][UPF][PFCP][LAddr:192.168.18.149:8805] handleAssociationSetupRequest
2025-10-30T01:54:26.999101771-07:00 [INFO][UPF][PFCP][LAddr:192.168.18.149:8805][CPNodeID:192.168.18.145] New node
2025-10-30T01:54:26.999569915-07:00 [INFO][UPF][PFCP][LAddr:192.168.18.149:8805] handleHeartbeatRequest
2025-10-30T01:54:30.796674020-07:00 [INFO][UPF][Main] [UPF] 🔥 Added iptables rule: DROP all TCP traffic to 192.168.211.129:80
Rule: -A FORWARD -d 192.168.211.129/32 -p tcp -m tcp --dport 80 -j DROP
2025-10-30T01:54:37.002125863-07:00 [INFO][UPF][PFCP][LAddr:192.168.18.149:8805] handleHeartbeatRequest
2025-10-30T01:54:40.471028569-07:00 [INFO][UPF][PFCP][LAddr:192.168.18.149:8805] handleSessionDeletionRequest
2025-10-30T01:54:40.471084314-07:00 [ERRO][UPF][PFCP][LAddr:192.168.18.149:8805] handleSessionDeletionRequest: Sess: inv
eid:0
2025-10-30T01:54:40.515928664-07:00 [INFO][UPF][PFCP][LAddr:192.168.18.149:8805] handleSessionEstablishmentRequest
2025-10-30T01:54:40.515966678-07:00 [INFO][UPF][PFCP][LAddr:192.168.18.149:8805][CPNodeID:192.168.18.145][CPSEID:0x3][UPSEID:0x
1] New session
2025-10-30T01:54:40.517432133-07:00 [INFO][UPF][Perio] recv event[TYPE_PERIO_ADD][{eType:1 lSeid:1 urrid:1 period:30000000000}]
2025[illegible]517451[illegible] [INFO][UPF][P[illegible]ticker [illegible]
2025[illegible] [INFO][UPF][P[illegible]recv event[[illegible]{eType[illegible]lSeid[illegible]:2 period:30000000000}]
2025-10-30T01:54:40.527629764-07:00 [INFO][UPF][PFCP][LAddr:192.168.18.149:8805] handleSessionModificationRequest
2025-10-30T01:54:47.006242741-07:00 [INFO][UPF][PFCP][LAddr:192.168.18.149:8805] handleHeartbeatRequest
2025[illegible]PFCP][LAddr:192.168.18.149:8805] handleHeartbeatRequest
2025[illegible]ain] [UPF] 🔥 Added iptables rule: DROP all TCP traffic to 192.168.211.129:80
Rule: -A FORWARD -d 192.168.211.129/32 -p tcp -m tcp --dport 80 -j ACCEPT
```

```
iotlab@gnb1:~$ sudo iptables -F FORWARD
iotlab@gnb1:~$ sudo iptables -S FORWARD
-P FORWARD ACCEPT
iotlab@gnb1:~$ sudo iptables -S FORWARD
-P FORWARD ACCEPT
-A FORWARD -d 192.168.211.129/32 -p tcp -m tcp --dport 80 -j DROP
iotlab@gnb1:~$ sudo iptables -S FORWARD
```

```
gNB1 & UE: distance=95 m | dbm=-90
[gNB] A3 Event detected from UE (sti=0x31d170d3b5c0cb8)
[gNB] Sending HandoverRequired to AMF...
[NGAP] Triggered SendHandoverRequired()
[NGAP] Sent simplified HandoverRequired NGAP PDU (clientId=2)
56. Position: gNB1=(0,0,0), UE=(56,56,56)
gNB1 & UE: distance=96 m | dbm=-87
[gNB] A3 Event detected from UE (sti=0x31d170d3b5c0cb8)
[gNB] Sending HandoverRequired to AMF...
[NGAP] Triggered SendHandoverRequired()
[NGAP] Sent simplified HandoverRequired NGAP PDU (clientId=2)
57. Position: gNB1=(0,0,0), UE=(57,57,57)
gNB1 & UE: distance=98 m | dbm=-90
[gNB] A3 Event detected from UE (sti=0x31d170d3b5c0cb8)
[gNB] Sending HandoverRequired to AMF...
[NGAP] Triggered SendHandoverRequired()
[NGAP] Sent simplified HandoverRequired NGAP PDU (clientId=2)
58. Position: gNB1=(0,0,0), UE=(58,58,58)
gNB1 & UE: distance=100 m | dbm=-93
[gNB] A3 Event detected from UE (sti=0x31d170d3b5c0cb8)
[gNB] Sending HandoverRequired to AMF...
[NGAP] Triggered SendHandoverRequired()
[NGAP] Sent simplified HandoverRequired NGAP PDU (clientId=2)
```

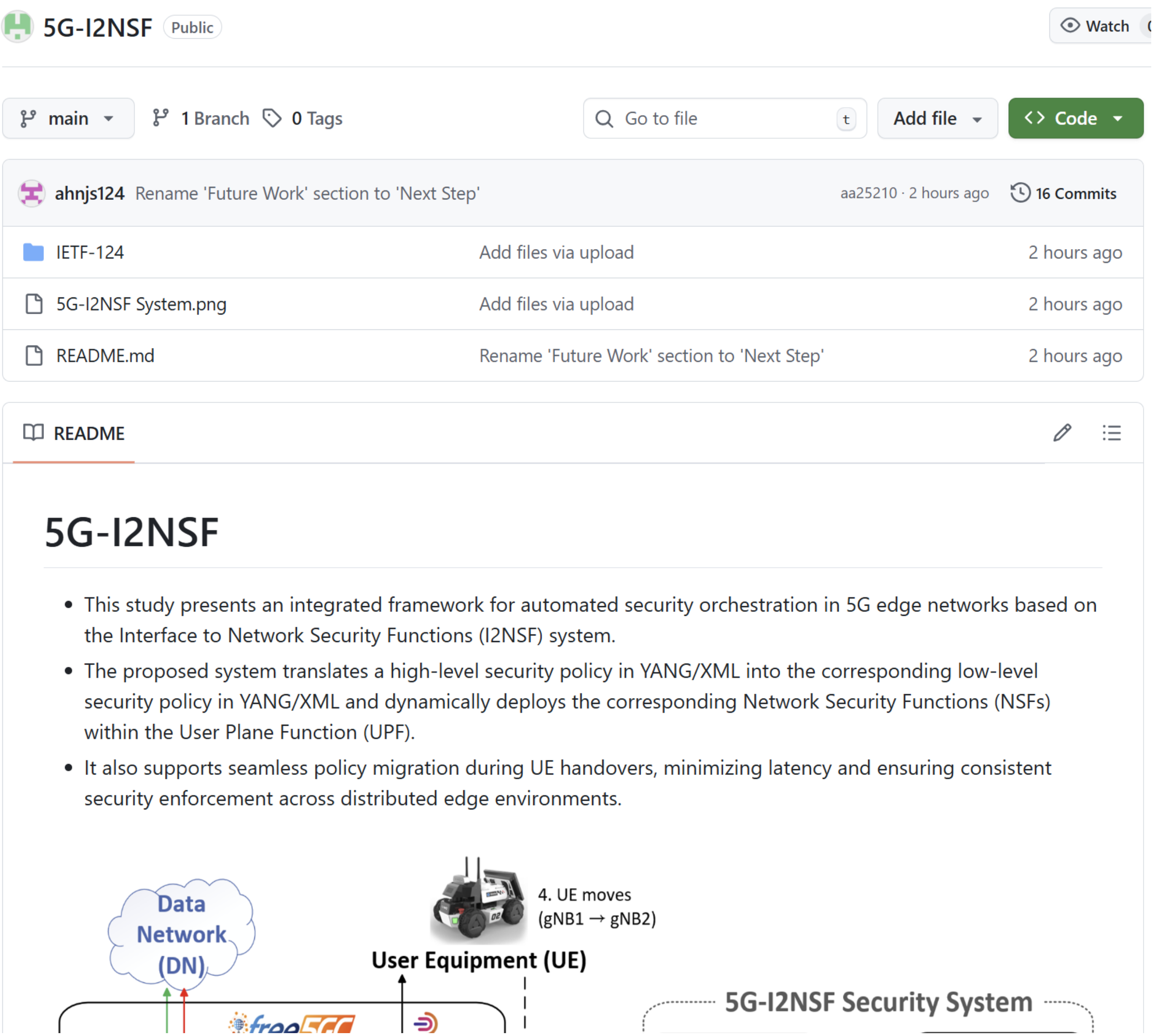
Created NSF (Firewall) on UPF2

Create NSF on UPF2 based on the migrated policy

Checking A3 Event (Handover) of UE

Open-Source Project for 5G-I2NSF

[URL] https://github.com/jaehoonpauljeong/5G-I2NSF



ahnjs124 Rename 'Future Work' section to 'Next Step' aa25210 · 2 hours ago 16 Commits

IETF-124	Add files via upload	2 hours ago
5G-I2NSF System.png	Add files via upload	2 hours ago
README.md	Rename 'Future Work' section to 'Next Step'	2 hours ago

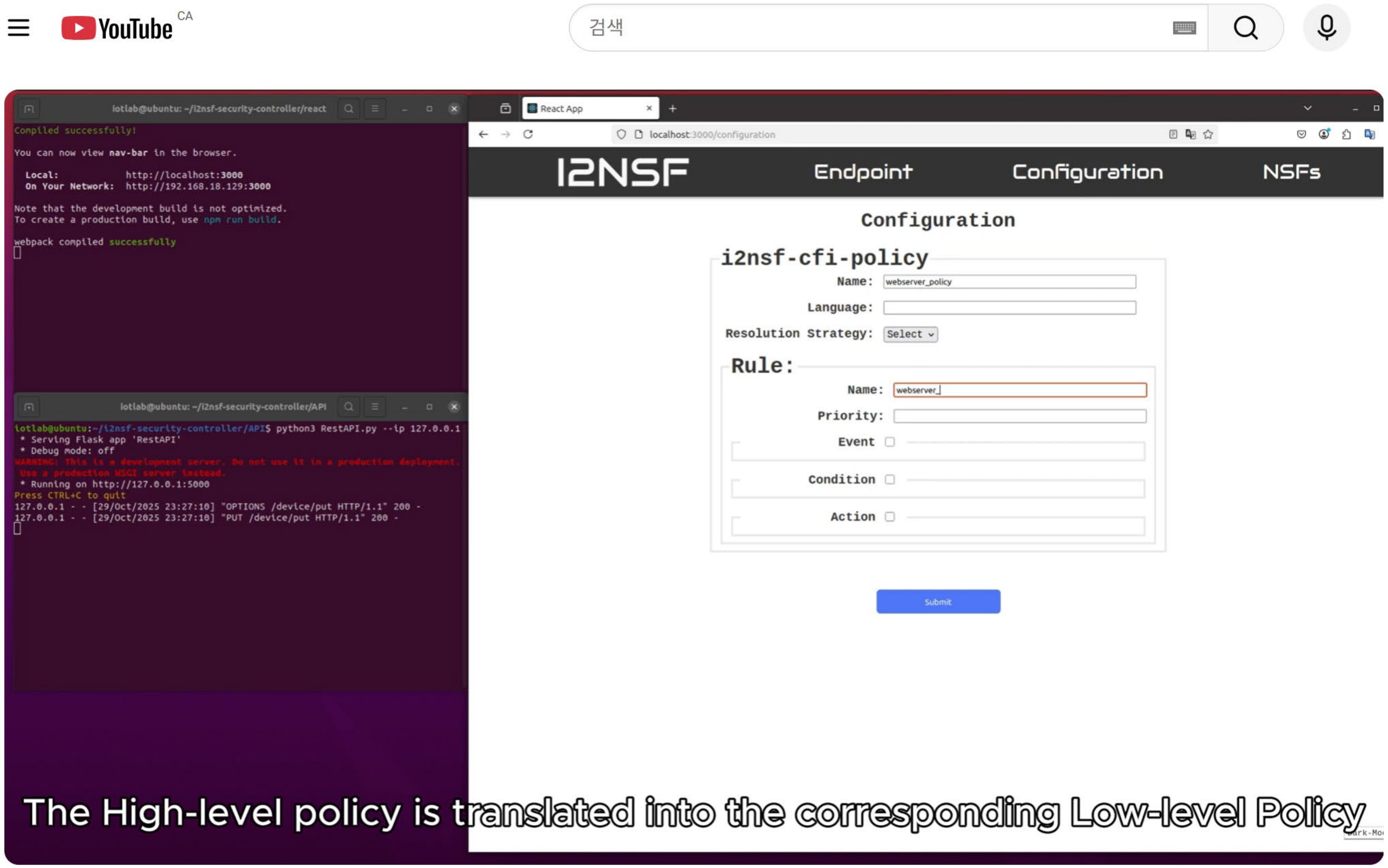
README

5G-I2NSF

- This study presents an integrated framework for automated security orchestration in 5G edge networks based on the Interface to Network Security Functions (I2NSF) system.
- The proposed system translates a high-level security policy in YANG/XML into the corresponding low-level security policy in YANG/XML and dynamically deploys the corresponding Network Security Functions (NSFs) within the User Plane Function (UPF).
- It also supports seamless policy migration during UE handovers, minimizing latency and ensuring consistent security enforcement across distributed edge environments.

Demonstration Video Clip for 5G-I2NSF

[URL] https://www.youtube.com/watch?v=YxX7MGFy65E



iotlab@ubuntu: ~/i2nsf-security-controller/react

```
Compiled successfully!

You can now view nav-bar in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.18.129:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

iotlab@ubuntu: ~/i2nsf-security-controller/API

```
iotlab@ubuntu:~/i2nsf-security-controller/API$ python3 RestAPI.py --ip 127.0.0.1
 * Serving Flask app 'RestAPI'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
 Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [29/Oct/2025 23:27:10] "OPTIONS /device/put HTTP/1.1" 200 -
127.0.0.1 - - [29/Oct/2025 23:27:10] "PUT /device/put HTTP/1.1" 200 -
```

Configuration

i2nsf-cfi-policy

Name: webserver_policy

Language:

Resolution Strategy: Select

Rule:

Name: webserver_

Priority:

Event □

Condition □

Action □

The High-level policy is translated into the corresponding Low-level Policy

IETF-124 5G-I2NSF System for Integrated Security Services in 5G Networks

Next Steps

- We will convert YANG/XML data models to YAML data models for I2NSF.

- We will design and implement the 5G protocol procedure for I2NSF considering the interaction among UE, AMF, SMF, SCF, and NSFs.
  - Under handover scenario, a security policy will migrate from a UPF

- We will reflect our hackathon experience on our draft:
  - https://datatracker.ietf.org/doc/draft-ahn-nmrg-5g-security-i2nsf-framework/

5G-I2NSF Hackathon Team

Professors:

- Jaehoon Paul Jeong (SKKU)
- Younghan Kim (SSU)
- Yong-Geun Hong (DJU)
- Joo-Sang Youn (DEU)

Researchers:

- Jung-Soo Park (ETRI)
- Yunchul Choi (ETRI)

Students:

- Yoseop Ahn (SKKU)
- Jiwon Suh (SKKU)
- Jiwon Yoo (SKKU)

Hackathon Team Photo

Appendix

Why do we integrate I2NSF into 5G Networks?

- The 5G network introduces massive device connectivity, edge computing, and frequent mobility, including UE handovers.

- Traditional I2NSF deployments rely on centralized cloud-based architectures, which are not suitable for latency-sensitive edge environments. During UE handovers between gNBs, the system must reconnect to a central server, causing additional latency and reducing responsiveness.

- Therefore, we propose an integrated architecture that combines I2NSF-based components (e.g., IUF, SCF, SDAF, and NSF) and the 5G service architecture to enable intent-based security management within the 5G Core.

- The proposed system supports <u>seamless policy migration during UE handovers</u>, minimizing latency while ensuring consistent security enforcement across distributed edge environments.